ABERYSTWYTH UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

# Neuro-Evolution and Deep-Learning for Autonomous Vision based Road-Following

*Author*
Aparajit NARAYAN

*Supervisor*
Dr. Frédéric LABROSSE

12 April 2018

**Neuro-Evolution and Deep-Learning for Autonomous Vision based Road-Following**

by

Aparajit Narayan

## Abstract

The ability to robustly detect and follow roads, irrespective of their type and prevailing environmental conditions is a crucial component of the control system of autonomous vehicles. Navigation in scenarios where the boundaries between road and non-road areas are poorly delineated, extreme lighting conditions such as shadows and reflections, sudden changes in road composition are some instances of challenging conditions which makes road-following a non-trivial problem from a computer vision perspective. A review of the state of the art suggests that despite being the focus of a vast body of works, there is a need to explore alternate approaches for developing a control framework that can be applied across the variety of operational scenarios. In this thesis two solutions based on different implementations of artificial neural networks are proposed for robust and generalised autonomous driving. The first solution presented in Chapter 3, is an integrated sensorimotor controller directly controlling the designated mobile platform, that is trained using the principles of evolutionary robotics (neuro-evolution) on a set of virtual roads. The controller is able to adapt to new road environments by dynamically adjusting its own perception of the environment to extract desired regularities. Analysis into the behaviour of this network suggests that despite a very low resolution visual input, it is able to extract discernible cues from the environment through complex patterns of oscillations which dynamically alter the composition of colour channels composing its final input vector. As opposed to this active vision system, the solution in Chapter 4 is based on the use of convolutional networks that predict the position and width of the road in the input image plane. These passive vision networks are trained to learn features global to road-environments on a dataset that encapsulates varied operational conditions. Both solutions are extensively evaluated with a number of different colour models in virtual and real-world tests involving a Pioneer 3-AT mobile robot. Results from driving trials on a set of common road segments suggest the former dynamic active vision controller performs better than a control system that makes navigation decisions based on road position predictions from the deep convolutional network models. The two controllers based on neural models with differing architectures and training schemes were shown to be able to generalise to noisy real world road environments, much different from those used during training. Besides a providing comparative evaluation of the two approaches, we also discuss future directions of research through which the principles of neuro-evolution (evolutionary robotics) and deep-learning can be integrated into a single control structure.

# Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

**Aparajit Narayan**
*05.05.2018*

STATEMENT 1

This work is the result of my own investigations, except where otherwise stated. Where correction services have been used, the extent and nature of the correction is clealry marked in a footnote(s). Other sources are acknowledged (e.g by footnotes giving explicit refernces). A bibliography is appended.

**Aparajit Narayan**
*05.05.2018*

STATEMENT 2

I hereby give consent for my work, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

**Aparajit Narayan**
*05.05.2018*

# Acknowledgments

I would firstly like to thank my supervisor Dr. Frédéric Labrosse and previous supervisor Dr Elio Tuci for their continued support throughout my PhD and invaluable contribution to the work presented in this thesis. I am also grateful to Fujitsu and HPC-Wales for funding this project and providing the computational resources that made our experiments feasible. I would like to mention my colleagues in the Intelligent Robotics office, fellow PhD students, lecturers and support staff in the computer science department of Aberystwyth University who have all been incredibly helpful and supportive of my research. Last but not the least, my friends, family and football teammates have also provided a lot of encouragement and motivation over the past four years for which I am very grateful.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Autonomous or ‘self-driving’ vehicles, with their potential to revolutionize personal and commercial transportation have been the focus of researchers for a number of decades. From systems in the early 80s which could only travel limited distances in controlled scenarios, to recent developments by corporations such as Google and Tesla which have demonstrated successful driving over thousands of kilometers in cities and highways; the technology of autonomous driving has come a long way. Indeed automotive manufacturers have begun incorporating semi-autonomous driver-assistance and safety features like lane-detection, self-parking, ‘auto-pilot’ mode, obstacle avoidance in current models. This trend is only set to accelerate over the coming years and besides individual cars, numerous possible applications can be found in domains of public-transport, supply-chain logistics, automated surveying etc. Despite recent advances, there are still a number of technological hurdles for the large scale implementation of fully autonomous driving technology can be realised. Road safety and public perception implications mean commercial self-driving vehicles must incorporate mechanisms for robust operation in varied dynamic settings, including scenarios that may not have been considered during the design phase. Despite recent advances, there are still a number of technological hurdles for the large scale implementation of fully autonomous driving technology can be realised. Road safety and public perception implications mean commercial self-driving vehicles must incorporate mechanisms for robust operation in varied dynamic settings, including scenarios that may not have been considered during the design phase.

## 1.1 Road-Following: A non-trivial problem

Depending on the nature of application and/or environment, driver-less vehicles may incorporate a range of functionalities such as road-following, decisions on intersections, static and dynamic obstacle detection/avoidance, traffic symbol detection and route planning. These can be considered to be more or less separate fields of research with advanced state-of-the-art associated with each of them. In this thesis, the focus is solely on the road-following, i.e. the ability to drive while keeping within the limits of the road which is essential to any automatic driving system. While this may appear to be a somewhat simplistic problem for human eyes, its complexity becomes apparent if the entire spectrum of possible roads and operational scenarios that exist for a autonomous vehicle is taken into account. The definition of a 'road' is in itself somewhat subjective and can be applicable for a range of environments from urban roads and motorways to poorly demarcated unpaved tracks. We attempt to define a road as 'a stretch of traversable surface, wide enough for a reasonably sized vehicle or mobile robot to travel and visually distinguishable from its surroundings (through either one or multiple features)'. It is acknowledged that for truly autonomous driving, especially in more unstructured and off-road conditions, the concept of traversibility is a more complex issue. There is a line of work (e.g. [61]) that considers the estimation of traversable surfaces an 'affordance'; which is learned by the robots interactive exploration of the environment through which it is able to determine which surfaces/textures it can 'afford' to drive on. This is inspired by the 'ecological approach to visual perception' published in [39] which espouses the importance of explorations of the environment during the learning phase of biological organisms to extract structural regularities that enables them to establish the nature of their relationship with the environment. However we consider this an issue beyond the remit of this thesis and simplify the problem with the assumption that all roads considered are 'traversable' by the targeted mobile platform.

A truly autonomous driving controller must address this issue of environmental variability by being able to traverse roads irrespective of the surface composition (i.e. asphalt, concrete, dirt etc.) and surroundings (i.e the non-road area). Besides this it should be able

to adapt to dynamic variations in real-time as the robot progresses along a particular road or path. Noise due to weather conditions (shadows, puddles, bright reflective spots etc.), poorly delineated road edges, lack of discernible information in the input feature-space, presence of pedestrians and other vehicles etc. are some of the challenges in the way of developing such controllers. Other factors that may influence their practical viability include hardware requirements (sensory and computational), execution time of input-output cycles and ease of integration as part of a larger autonomous-driving control-stack.

## 1.2 Project Scope

This project was funded by Fujitsu and HPC-Wales to commence in December 2013, when deep and recurrent neural networks were becoming an increasingly popular choice for a number of problem domains such as object recognition and natural language processing. The overall research agenda was to explore the application of neural-network based models for delivering a vision based road-following controller that could be applied across the spectrum of possible road environments, especially taking into consideration challenging conditions such as unpaved/delineated boundaries, dynamic lighting variations and changes in the colour composition of roads. Access to high performance computing resources was provided to ensure common problems associated with limitations in training time and data with these models could be navigated to produce generalised solutions. Whilst other sensor cues apart from image pixels could be have been considered, road-following was assumed to be a pure vision problem so that the methodologies developed for this project could be generic enough for application in other computer vision tasks which similarly require differentiation between multiple areas in the image plane.

## 1.3 Thesis Outline and Contributions

This thesis proposes two solutions based on differing implementations of neural network topologies, towards developing a robust and generalised road-following controller. The first solution presented in Chapter 3 is a recurrent neural network receiving camera images as

input to control a mobile robot by setting its wheel speeds as well as control the colour composition of the visual input. The network is trained by means of a genetic algorithm wherein populations of genotypes (each representing a set of network weights) are evaluated for their 'fitness' on a road-simulator specifically designed to encourage emergence of adaptive sensorimotor control. In contrast to this dynamic 'active-vision approach' (i.e. ability to alter its own perception of the visual scene); the solution in Chapter 4 proposes the use of deep neural networks to predict the position and width of roads present in a camera image. Convolutional neural networks (CNNs) are first trained/evaluated 'off-line' on a dataset of annotated road images, after which detection outputs from selected networks are used to control a mobile robot in real-world trials. Under this approach we implement a static-controller forgoing system dynamics (due to the lack of recurrent/memory nodes); instead relying on the neural-architectures ability to decompose the input space into global hierarchical representations that are robust to environmental variability and disturbances. Refer to Section 2.3 for a more detailed discussion on the theoretical principles behind both approaches. A comparative evaluation of the two solutions along with proposed improvements and possible avenues for their integration into a single control-framework is presented in Chapter 5.

A review of key road-following literature is presented in the following Chapter 2. Here we also discuss the necessity to explore alternate and novel approaches that may be able to address/circumvent limitations of the methodologies developed thus far. Arising from this need to delve beyond standard algorithmic approaches to road-following (discussed in Chapter 2), we look to extend the use of connectionist models (i.e. neuro-controllers) in road-following problem domain. A review of neural-network and machine-learning based approaches (in Chapter 2.2) shows state-of-the-art works often lack the adaptivity required for achieving higher degrees autonomy desired by us. While neural-networks have been successfully used for autonomous driving applications they are often restricted to specific types of roads and lack the mechanisms to deal with dynamic variations, even along the specific road types they are designated for. To address these limitations we develop the two neural-network based controllers, using inspiration from the separate fields of evolutionary robotics and deep-learning respectively. Both approaches offer their own unique set of

advantages that in theory can lead to robust and generalised road-following behaviour, beyond what is achievable from the implementations reviewed in Chapter 2.2. The dynamic active vision controller attempts to levarage the neuro-controllers unhindered explorations of simulated roads during its learning phase to 'evolve' perceptual strategies that can be transferred to challenging real-world conditions. On the other hand, with the 'deep convolutional network' which unlike the former approach doesn't directly control the robot's motion; we forgo the benefits of active vision and evolutionary embodied learning (discussed in Chapter 2.3) for the potential of learning globalised high-level road features. We also investigate the impact of alternate representations of colour (besides the standard RGB model) on the performance of both controllers. In both cases an extensive and systematic analysis of the characteristics of the evolved/trained solutions has been carried out. Evaluation of both controllers on a set of 5 outdoor paths/road-segments, demonstrate both controllers can successfully guide a mobile robot on roads different from those encountered during training besides providing means for their direct quantitative comparison. In addition, although current technical know-how and computational resources available led us to investigate two separate approaches; this work has thrown insight into strategies wherein the advantages offered by each of them can be utilised in conjunction during road-following trials (discussed in Chapter 5). After an evaluation of both approaches, we advocate bridging the fields of evolutionary robotics and deep-learning for systems that are closer to the performance and our current understanding of biological vision and the learning apparatus required for its development.

The primary contributions of the work carried out during this project are summarised below:

- Continuous time recurrent neural networks (CTRNN) with low resolution visual fields were trained and evaluated to control a virtual robot using the principles of evolutionary robotics (neuro-evolution) on a set of simulated (computer rendered) road sequences. These networks learnt to dynamically control the composition/mixing of colour channels formulating their own input vector by activating designated feedback nodes controlling this ratio. They were ported onto a real mobile robot platform (Pioneer 3-AT) and successful adapt to noisy environments with complexities

not encountered during training (moving pedestrians, varying colour composition of road/non-road areas, extreme lighting conditions). Analysis into the behaviour of these networks shows their colour feedback nodes exhibiting a series of complex oscillatory patterns through which desired visual cues are extracted.

- Deep convolutional networks of different architectures were trained to identify the position of the road on a dataset of images representative of different environmental conditions. These networks could generalise to unseen road images, including those from environments not present in the training set, with accuracies comparable to a bench-mark computer vision based method. Road detection predictions from these deep CNNs were also shown to successfully guide a Pioneer 3-AT robot in real-time on a number of different indoor and outdoor road segments, albiet with lesser accuracy when compared to the recurrent networks with active colour perception mentioned above.

- The use of colour representation besides the standard RGB model improves the performance of both approaches. The CTRNN models trained with RGB images were found to perform better when alternate colour models were used. However contrary to this CNN models only displayed acceptable degrees of accuracy with the same colour model that was used for training set images.

Refer to Chapter 5 for further discussion on these inferences derived from the series of experiments presented in Chapters 3 and 4. Experiments presented in this thesis have been published as part of international conference proceedings as well as in the journal 'Neurocomputing'. The details of these publications are listed below:

- A dynamic colour perception system for autonomous robot navigation on unmarked roads. *Neurocomputing Vol 275: 2251-2263*

- Simulated Road Following Using Neuroevolution. *ALIA 2014: 17-30*

- An active vision approach to the road following problem. *IROS 2015: 4650-4655*

- Road detection using convolutional neural networks. *ECAL 2017: 314-321*

# Chapter 2

# Motivation for methodologies implemented in this work

From the previous chapter it can be seen that for a truly autonomous driving control-stack, it is crucial to develop a solution to detect and follow roads across all environmental conditions. This has indeed been an important issue for researchers across decades and led a wide array of proposed solutions, some effectively demonstrating automatic driving in challenging conditions. A review covering the entire spectrum of literature relevant to road-following is beyond the scope of this thesis; however important works which have contributed to state-of-the art improvements and development of promising techniques in this field have been presented in this section. Most works rely on modifications to standard signal processing, statistical and probabilistic methods while others attempt to develop control solutions based on machine learning and artificial neural network models.

## 2.1 Road-Following: Key design based approaches

There is a also a degree of diversity in the wide body of literature that discusses 'engineered solutions to autonomous road-following' in terms of models used for representing the road/non-road characteristics, targeted operational environment, the types of sensor inputs relied on and available/targeted computational capabilities (e.g. some solutions target low-power embedded chips while others are dependent on more expensive parallel compu-

tational resources). A lot of especially early works in this area of research address structured environments such as highway lanes and well-delineated urban roads. The majority of these approaches are reliant heavily on feature types (e.g., road markings [29, 62]) that are often missing or not detectable from visual scenes (outside the specific type of environment they have been designed for). The authors of [29] have published a series of experiments in road-following for more than a decade over the course of which they developed and modified a complete autonomous driving control-stack. The road-detection strategies based on tracking road-edge features used by them contributed to successful high-speed driving (100 km/ph) over long courses of highway roads (see [30]). However this approach does not hold for unstructured roads where it becomes difficult to maintain a lock on these features. To overcome this the authors introduced a method based on having pre-defined global assumptions of the road-structure and using this to aid in the search for helpful visual cues from the road boundaries. Similarly in [62] explicit features like painted white and yellow lines are tracked by the detection algorithm and these are transferred to a road model that makes strict assumptions about geometrical features such as road curvature that apply only to highway lanes. The search for discriminative features is thus contained within a limited range of area within the raw image input.

While such methods relying on pre-defined or fixed features can prove to be robust against local variations (e.g. shadows, rain), algorithms addressing unstructured roads need to track a specific set of features that are universal to road-following scenarios and/or be able to dynamically extract the correct features for every environment they are operated in. Two pioneering works which have proved successful in demonstrating detection and driving in unstrcutred roads were the SCARF (Supervised Classification Applied to Road Following) [24] and its predecessor the UNSCARF (Unsupervised Classification Applied to Road Following) [23] system. A number of subsequent solutions (e.g. [95], [123], [112] etc.) have been inspired by and tried to improve upon the core methodologies used in [24] and [23]. Road detection in these works is based purely on vision and relies on essentially differentiating between the colour distributions in the road and non-road parts of the raw input image. In [24] the position of the road is first detected based on manual user-inputs or through a unsupervised initialization scheme detailed in [23]. An essential contribution of

this work is that no prior knowledge of the environment, additional sensor cues or maps are required to detect and maintain a lock on the road. The authors also advocate using a dual camera system to increase the dynamic range of the raw inputs and capture informational essential for dealing with extreme lighting conditions. A pre-processing step based on combining subsampling and pixel averaging techniques is used for dimmensionality reduction to reduce computational load on subsequent steps as well as to remove noise inherent to the image capturing process. After the initial road/non-road areas have been allocated pixels belonging to both these regions are clustered using a standard nearest-mean based technique into four sets each. These sets are then used to form statistical models based on their respective colour distribution properties. In subsequent frames new resampled pixels are classified into one of the road or non-road classes based on the probabilities (computed using Bayesian principles) of them belonging to the associated 'cluster models'. This classification is then projected onto a road model represented geometrically in the shape of an isosceles triangle. It should be noted that the choice of four models to represent the road/non-road areas respectively was an arbitrary decision based on empirical observations by the authors. While this choice was justified for the subset of environments evaluated by the authors, it cannot be said to be robust enough to account for the vast range of possible road environments. Because of this static representation of colour features (either as statistical or Gaussian based models), the performance of SCARF (and similar approaches) comes into question when the distributions of colours in road/non-road areas are similar to each other and pixels belonging to either of these regions are mis-classified. Another potential cause of failure is rapid dynamic changes in the road. Not processing and integrating these changes into the models results in redundant colour information being stored in them. The geometric representation of the road is also built upon structural assumptions that it is of constant width and locally linear. While these assumptions may hold for the majority of roads, there are cases (especially in severely unstructured environments) where a more complex geometrical model is required to account for sharp bends and changing changing road-widths. Similarly the UNSCARF [23] approach was also limited by this simplistic geometrical model.

The UNSCARF [23] algorithm first extracts the road edges and then matches that to the

best fitting road shape (same as that in [24]) by minimizing a distance function against a set of predefined model interpolations. Each pixel is represented in terms of a five dimensional feature vector comprised of the R,G,B color channels and its coordinates in image space. For the edge extracting procedure, pixels (represented as a feature set) sampled from each new frame are clustered in an unsupervised manner into a number of arbitrary classes, the number depending on the individual frame and its colour distribution characteristics. After every pixel has been assigned a class label, the less dominant label names are removed and edges on the borders of these remaining clusters of pixels are extracted to be processed by the road matching algorithm. One of the key benefits of this approach was that no manual labelling or restrictions on the vehicle's relative position on the road were required. The authors however relied on extracting discernible road/non-road clusters from the R, G and B colour channels which could lead to failures under extreme illumination conditions (e.g shadows, puddles). While they propose the use of an alternate colour representation such as the HSI/HSV (Hue Saturation Intensity/Value) model, it was not included in their implementation due to image processing constraints.

In the work presented in [112] normalized R,G and B values are used to reduce effects of illumination noise. Multiple histogram based colour models (arbitrarily set to four) are used to represent the road. At each frame the authors produce a histogram based on the simplistic but relatively robust assumption that a small rectangular region, positioned ahead of the camera in the image plane, contains road pixels. This histogram is then compared to existing models and replaces the one that is most similar to it. However in the scenario that the difference between the newly processed histogram model and each existing model exceeds a predetermined threshold, the algorithm enters an iterative learning mode for a succession of frames. At the end of this procedure one of the existing models is completely removed and is replaced with the newly sampled model. The non-road areas are represented by a singular histogram model which is updated by incrementally integrating new histograms (formed by randomly sampling presumed non-road areas) at each step. Another key aspect of this work is the application of temporal fusion, wherein segmentation results from multiple frames are integrated to negate the effect of possible errors which could be present in individual frames. [123] also relied on adaptive construction of 3D histograms using the

RGB colour model. The input image is captured by a panoramic camera (for a wide field of view) and a process based on maximizing a fitness function is used to best fit probable road-shapes to a spherical geometric model.

A number of important segmentation based methods such as [58], [99], [74] rely on using Mixture of Gaussian (MoG) models instead of histograms. 3D histograms and Guassian mixture models provide an almost comparable degree of performance for region-based segmentation applications such as road detection (see [67]). The detection algorithm presented in [99] was part of a larger autonomous driving system used to control the robot that won the DARPA grand challenge in 2006 by driving through a highly delineated desert course. Readings from a laser range finder are first used to extract traverse-able areas ahead of the robot in form of a quadrilateral shape projected onto the image plane. This driveable part of the road is represented by multiple Gaussian distributions modelling the RGB colour characterstics of pixels within it. The authors highlight the need to adapt to both incremental dynamic variations as well as more drastic changes such as the robot moving onto a completely new road type (e.g. from dirt to asphalt). To achieve this newly sampled pixels at each frame are used to form a number of temporary Gaussians, which are then compared to existing distributions based on the Mahalanobis distance between them. Depending on an predetermined error threshold, new pixel information is either incorporated into the existing model or the model itself is updated by adding/discarding select Gaussians. A decision tree based classification method [28] relying on extracting both road and non-road areas was also tested as a benchmark. Because of the complex color and structural characteristics of non-road regions, the Gaussian based approach (relying only on road pixels) displayed better performance in the tested routes. [74] also uses a similar Gaussian based segmentation approach for operation in unstructured desert roads. Pixels from predicted road and non-road areas are sampled to maintain Mixture of Gaussian models from both these areas. The authors advocate the use of the Expectation Maximization Algorithm [84] (over Baysian methods) for classifying pixels into either of the road/non-road models.

Besides adaptive pixel modeling and classification techniques, it is also important to consider the integrity of the input sources themselves. Using channels that do not contain enough discriminating information is a common drawback in road-following/detection

works. For example, because of its inherently fused representation of chrominance and luminance, the commonly used RGB colour model is often poor at extracting discernible features in shadows (and other complex lighting conditions). In [5] (another histogram based solution) the authors first transform the raw image into an illuminant-invariant color space to reduce the effect of shadows. This proposed feature representation outperformed a baseline algorithm (using the HSI colour space) on images corresponding to a variety of dynamic lighting conditions on urban roads. There are cases however (e.g. off-road dirt paths) where texture and/or depth based features are more useful than those just extracted from colour models.

An 'all-terrain' path following system described in [3] uses features derived from applying a Walsh-Hadamard filter-bank [48] to model blocks of pixels in the input image. In conjunction with this texture based block classification system, a second detection module based on extracting shape regularities through optic flow was also implemented. This module worked under the premise of strict assumptions about the robot's relative position on the road and the geometrical shape of the road itself, and was designed for scenarios (e.g complex lighting conditions) where the accuracy of the texture based classifier was not reliable. The authors of [97] also devise a system that only uses texture regularities for rural/unstructured roads. This approach is especially beneficial in scnearios of almost no colour contrast (desert/snow tracks), where global regularities (e .g. vehicle trails) need to be exploited to stay on course. A set of Gabor filter-wavelets [35] is applied over the input image to extract correlations with orientations corresponding to a predefined set of angles. These 'dominant' orientations are voted into a global vanishing point (which is then tracked temporally) to provide trajectories/cues in aid of vehicle navigation. The process of selecting one vanishing point in the scene from this principle of dominant features can indeed work in very isolated and unstructured rural roads. However the presence of objects and buildings in the environment can also result in false voting for this vanishing point. [96]) also uses texture cues, but in conjunction with colour and 3D laser data to provide a richer and more robust representation of the road.

This concept of fusing colour and depth or 3D information has also been employed in a number of other important works such as [16], [80], [104], [99], [77] and [92]. The solution

described in [104] uses a pair of stereo-vision cameras (unlike [99] which employed laser range finders) for free-space estimation in front of the vehicle. It then uses an unsupervised clustering algorithm (ISODATA [54]) to segment regions of the image based on the Hue and Saturation (discarding luminance from the HSV colour model) distribution of pixels within these regions. Based on the assumption that the road can be geometrically modelled by an isosceles triangle, each of these clusters is further classified into two overall road and non-road regions.

Another instance of integrating information from different feature spaces is [81]; where an autonomous mobile robot is shown to navigate stretches of highly unstructured and cluttered environments (e.g. dirt and forest paths) using particle filters for estimating the road-curvature. Particle filters (see [8] for more details) are an approach used for dynamic state estimation, i.e, predicting state of a system based on information about the current state. Weighted probabilities are set for likely candidates (probable road regions in the case of autonomous driving) in the output-parameter's search space. An interative search process follows this, through which the distribution of probable road shapes is narrowed down to a range within acceptable error limits. Initial work for road-detection using particle filters ([105], [7]) focused only on well-demarcated roads and relied on tracking lane-marking cues. Subsequently the benefits of cue fusion were utilized by other approaches (see [103], [77], [121], [81] and [121]) to successfully apply this estimation technique for noisy poorly defined road environments. In [81] the authors select the saturation channel (from the HSV colour model) and edge intensity and direction gradients extracted from RGB pixel values as features for the particle filter estimation. Constraints on the road width and location are further factors incorporated into the estimation procedure to prevent 'false particles' in areas too far away in the state-space. The authors model the road as segments of two clothoid spirals (also known as Euler spirals) to account for sharp curvatures often found in rural/forest roads (see [101] for more information on the use of clothoids to model road trajectories). Particle filters have demonstrated better performance (see [81] for detailed results) over other state estimation techniques such as Kalman filters (used in [119]) for tracking and navigating unmarked road environments. However the detection accuracy of such systems is affected by the variance of estimated particle distribution, which in turn

depends on the reliability of their extracted feature cues. In other words, estimation accuracy and uncertainty is tied to the degree of discernible information available in the selected feature space. For example, in [81] the choice of colour saturation values as one of the feature cues is based on the assumption that more colored areas in the image correspond to regions outside the road. This holds for certain select environments (due to the presence of vegetation etc.) where this work is evaluated in (e.g. forest paths with vegetation at the sides), but may not generalize across the broader range of road-following scenarios.

The authors of [91] also demonstrated successful driving in a variety of poorly demarcated road environments by using a shape-constrained geometric model of the road, rather than segmenting the entire image, along with a Gaussian model of the road color. The authors propose an adaptive method that uses the Mahalonobis distance between the color model of the road and pixels of the images. The geometrical model is fitted to the current image by minimising the Mahalanobis distance and maximising the width of the detected road. The accuracy of these color-based techniques to build a dynamic color model of the road is influenced by the number and type of color components used. The work described in [91] shows that the higher the variability in operating conditions, the more likely any fixed set of channels will fail in discriminating road from non-road areas. The authors conclude that the variability that can be faced by a robot that navigates a road is such that the road model needs to be constantly updated to respond to changes in illumination, surface and localized features such as puddles and shadows. Despite low error rates achieved by this 'adaptive statistical colour' (ASC) based method in online (real-time) and off-line (pre-captured) datasets, there were a number of failure cases and sequences of systematic detection offsets. Moreover, as the road model maintained by the method relies on colour characteristics of the previous frame, any sudden/drastic changes could potentially result in detection errors.

From this review of the commonly used road following/detection techniques it is clear that there has to be considerable forethought on part of the 'designer' to ensure robustness in both, the strategies for extracting features from raw inputs as well as the targeted features themselves. A hand designed controller would often reflect the designers own biases and fail to account for the sheer amount variability in operational environments. Works such

as [91] and [99], using the principles of maintaining a dynamic and adaptive representation of the road have indeed demonstrated a high degree of accuracy in challenging delineated conditions. However room for improvement exists as there are foreseeable scenarios where strong discriminative cues necessary for these methods to extract the road-shape may not exist. It is therefore necessary to consider road-following from the perspective of machine-learning and connectionist approaches. These techniques with their ability to 'learn-by-example' can in theory learn a global, generalized representation of roads, without prior human biases and restrictive assumptions that limit the performance of so many of 'hand-crafted' methods; although it is acknowledged that human choices still persist in the formulation of the training data and cost functions. Over the past few years major commercial self-driving platforms developed by companies (such as Google, Tesla, Uber etc.) have begun to incorporate machine-learning and neural architectures for most functionalities including road/lane detection. For road-following these platforms rely on a suite of LIDAR sensors and camera inputs (fused with GPS data) to maintain a map of the free-space around the vehicles proximity which enables them to make short-term and long-term navigation decisions. While early prototypes relied on variants of the segmentation based methods discussed in this section, the advent of deep learning has seen an increasing number of platforms relying to neural networks to extract lane markings. A case in point is the work carried out by NVIDIA which provides GPU enabled computational hardware (DRIVE PX) with deep neural networks pre-trained for various autonomous driving related tasks. NVIDIA has a partnership with TESLA to provide these systems optimised for deep-learning models for their autonomous vehicles. The deep CNN based methodology used by NVIDIA for road following ([13]) which has been shown to successfully drive in varied environments (including unpaved roads) is discussed in more detail in Section 2.3.2.

## 2.2 Machine-Learning and Neural Networks applied to Road-Following

Machine learning techniques and universal function approximators such as artificial neural networks have been proposed as a potentially effective solution to the problem of road detection in autonomous driving vehicles required to operate in noisy and highly variable real world conditions. A number of works that can be said to fall under the design-based approach covered in Section 2.1 (e.g. [3], [74]) apply machine learning classifiers in the image segmentation process to determine probability of a newly sampled pixel belonging to the road model.

In [124], image segmentation is carried out by a support vector machine (SVM) which classifies pixels as belonging to either road or non-road classes. Before the robot starts navigation, part of the road is selected from the initial frame. The pixels belonging to this region are used to train the SVM classifier. In the subsequent frames, additional road-pixels are added to the training samples (and old ones discarded), based on assumptions about the structure of the road. The classifier is thus continuously retrained and updated to adapt to changing properties of the road and recover from poor initial misclassification. However, classification errors can arise in complex environments and dynamic weather conditions when the pixels in the SVM training no longer represent the entire range of pixels on the road-surface.

An SSVM classifier (Strcutred Support Vector Machine [57]) was also used in [122], where it was trained to segment the image into two major groups (road and non-road). Three feature sets (Dense SIFT, Histogram of Oriented Gradients and Local Binary Patterns) form the basis of the SVM's training. This initial binary segmentation was then refined to a triangular road model which was then updated by new pixels sampled along the road. The authors acknowledge the complexity of devising a re-sampling procedure and feature representation that could make the SVM's predictions robust to sudden changes in illumination and colour properties. Another approach where the road model is constructed by online training is presented in [87]. Instead of SVMs a modified version of the connectionist Hebbian Learning approach (see [47]) is used to map a feature set generated

from applying GIST filters [90] over the image plane to an output space comprised of 15 streering control signals. Steering commands and frames collected from a initial supervised human driving is used to learn the Hebbian linkage matrix. The authors demonstrated better performance in off-line datasets and real-time driving in an indoors track over two other methods, Locally Weighted Projection Regression (evaluated in [88]) and Random and Random Decision Forests (see [14]). This approach was further improved upon in [89] by integrating speed control and unsupervised learning form performance feedback as the vehicle progressed. Despite successful driving in dynamic test tracks and the low computational resources needed for real-time performance, such online learning systems cannot be considered to be fully autonomous due to their reliance on the initial period of human intervention when training examples are generated.

Artificial Neural Networks with their ability to learn generalised cues from training examples without detailed task-specific constraints, can offer an attractive alternative to design based approaches. Most works (e.g. [118], [55], [96], [58]) applying neural networks for road-following use them as classifiers to assign high-level labels (e.g. road, vegetation etc.) to feature sets formed either individual or groups of pixels. A typical example of this approach is the neueral classifer proposed in [118] which uses for its input a feature descriptor formed from combining 3D-Laser data (providing free-space estimates) with color information from pixel clusters extracted using a modified Watershed Transform algorithm described in [9]. [55] which similar to [91] uses a trapezoidal shape to model the road structure uses Kalman Filtering techniques to maintain a limited region of interest in the image (to limit computational costs). RGB colour values from $3 \times 3$ blocks of pixels in this region are passed on to a neural network which after training on manual annotation from the first frame is able to perform a binary classification as to whether a block is on the road or not.

The ALVINN project described in [94] is one of the earliest examples of using neural networks to control an autonomous vehicle on real outdoor roads . The controller consisted of a 3 layered feed-forward neural network taking in a grey-scale image as input and outputting the needed turning in order for the vehicle to remain on the road. The system was first trained using back-propagation on data generated by a human controller navigating in a road-simulator (based on real road images) developed by the authors. Later networks

were trained "on-the-fly", using images and control commands from real-time driving on outdoor roads. After an initial period of training on a particular type of road-surface the network could take over control and drive autonomously, demonstrating that it had learned the correct mappings between input images and steering directions. However, one of the major limitations with the network was its inability to generalize across road environments different to what it had been trained for. To overcome this limitation, the same authors proposed a new modular architecture called MANIAC [59]. This new architecture consisted of several individual networks trained under different types of roads integrated in a modular structure, wherein activation from all the modules are combined to form the final output vector. While being capable of driving in multiple road-environments without needing to switch controllers, the system was still limited in the sense that accurately representing the entire range of possible road-types would require a progressively larger number of individually trained modules, and each time a new module was added to the system, the entire structure would require retraining.

In [116] a similar solution consisting of multiple network modules operating within a larger structure was proposed. The controller received a monochrome image which was first subjected to a feature extraction algorithm. The results of this was then passed on to a classifier stage (consisting of three concurrent self-organizing map modules) which predicted the robot's required trajectory (left, right or straight). The system was first trained and tested off-line on images from the *CMU (Carnegie Mellon University) Vision and Autonomous Systems Center*'s image database. It was also tested on a mobile robotic platform on outdoor roads after an initial training phase on images from these roads. In [102] another solution using multiple neural networks was proposed, to make the system applicable to a wider variety of roads. The networks were trained on images which were divided into blocks of $10 \times 10$ pixels. Each block was annotated as being either navigable (part of the road) or not-navigable (part of the non-road), and features extracted from a block formed the input to the network. The responses from multiple networks (different networks being fed a different set of features) were combined to predict the final classification results of blocks in new frames. The limitations of these methods that use neural networks trained for specific road types is that while they can adapt to further sections of already trained roads,

they cannot operate on previously unseen environments without further training.

A modified Radial Basis Function Network (RBFN [83]) instead of the conventional feed-forward multilayer preceptron neural network architecture was used in [98]. Two operating modes are explored, one based on off-line training on hours of manual driving data and another online unsupervised approach where the system generates its own training cues. For the former mode, an extensive period of driving is used to gather sequences of road images and manual driving commands which form training labels for the network. The manual controller for this needed to ensure a variety of different road conditions (different shapes and lighting conditions) were captured. Images were then clustered to form a set of road template shapes which formed the hidden layer of the RBFN. In the second method, only a few initial seed images are needed to form the road templates to which newly captured images are compared to. This unsupervised approach is proposed as a means for the vehicle to adapt to a rapid change in road conditions. However driving commands were observed to have large inaccuracies, especially in initial phases of learning. Moreover the system again relied on the premise of manual intervention triggering adaptation to a sudden change. While the off-line training approach can be robust to variations in roads it was trained for, it is poor at adapting to environments not encountered before. The authors used RGB greyscale values to represent pixel information, although their attempt to map individual pixels to higher level road shapes could have been improved if a more robust illuminant invariant representation of colour had been used.

## 2.3 Exploring Alternate Approaches to Road-Following

The works covered in Section 2.2 show that a neural-networks can be trained to demonstrate road detection/classification in challenging road conditions. However their accuracy and effectiveness depends on how similar the input cues generated from the operational environment are to those used for the learning and training phase. In other words, these feed-forward networks trained using regular gradient descent algorithms, become sensitive to cues that enable them to improve their performance in the environments chosen for testing and validation. They can perform reliably in untested environments only if they are ex-

posed to these previously learned regularities and these cues or regularities are appropriate for discrimination between the road and non-road regions. This phenomenon of 'overfitting' is a common problem for such neural network and other machine learning models (see [46]) and prevents them from being applied to a lot of similar problem domains where there is a high degree of variability and dynamism in the input space.

It should also be noted that there still exists a degree of human-bias with these methods, as the network's learning is closely tied to the training data selected by a human supervisor/designer. While it is hoped that the subset of images/data 'chosen' for training is varied enough to enable the networks learning to be generalized, this approach cannot be effective considering the uncertainty and unpredictability associated with such neural models operating in untested conditions. Moreover the labeled output data provided during training in cases where there the network directly maps the input vector to driving commands is a further limiting factor as these are usually generated by either manual annotation and/or capturing signals (steering, speed etc.) from driving runs performed by a human. The network's learning goal is to thus mimic the navigation strategies displayed by the 'human driver', which may not be sufficient to generate truly autonomous and adaptive behavior for operation in actual driving conditions. The training data generated using such an approach also does not account for situations in noisy environments when the network produces incorrect detection/steering outputs and requires (oftentimes sharp) adjustments to recover from these 'mistakes'. Sequences of human driving specifically recreating such near-failure conditions may add more variability to the data, but the costs (in terms of time, resources) associated with these procedures may outweigh any performance benefits they might have to offer.

Thus, more flexible and robust solutions are required to tackle generic, poorly delineated road-following scenarios. Motivated by these connectionist approaches ([94], [59]) and the promising results shown in [91] through exploration of alternate colour spaces, our objective is to design effective solutions to overcome the problem of dynamic local changes and global environmental variability. Two broadly separate approaches within the broader field of connectionist computational models, Evolutionary Embodied Active Vision and Deep Convolutional Neural Networks, are explored and used to implement the

methodologies presented in Chapters 3 and 4 respectively. Both approaches aim to use different facets of biological vision systems to achieve greater autonomy and generalisation for road-following. It should be understood however that the remit of this thesis does not provide for a detailed understanding of visual capabilities in natural organisms. It is indeed desirable to develop models that closely mirror vision capabilities of natural organisms (in terms of feature represntation, architecture, learning process etc), given the fact that the most advanced vision systems are those found in nature. This is however not possible to achieve with current technological and computational resources. For example, considering the number of synaptic connections as a measure, one of the largest artificial neural models to be implemented with one billion connections [70], represents only a small fraction of the 80-100 billion synaptic connections estimated in human brains. Researchers therefore only look at implementing very simplistic models of biological traits and behaviors to improve state of the art in their respective domains.

With the Evolutionary Embodied Active Vision approach the core principle is to use a relatively simplistic neural architecture that can dynamically vary its own perception of the environment. This can ensure that while the network is sensitive to only a small subset of important regularities that it extracts using its learned perceptual strategies to navigate roads in varying conditions. Another key aspect of this approach is having a unified sensorimotor loop, i.e. the network directly control the wheel speeds of the robot vehicle, which can result in more complex steering/navigation strategies (sharp turns, recovery from errors etc.) which are difficult to manually design. Because of this ability to alter its perception of the environment through self generated actions, the proposed neural network controller falls into the category of an 'active' vision system. A detailed understanding of the merits of such active over passive vision systems (which cannot alter their own view of the environment) is presented in [2]. The authors state that problems that are ill-posed and nonlinear for a 'passive observer' become well-posed and linear for an 'active observer'. Basic vision problems which require extracting shapes from contours/textures and inferring structural regularities in dynamic can be solved easier and more efficiently by an active observer. This principle is also employed by biological agents with relatively simplistic retinal and neural architectures (e.g. housefly) to intelligently navigate complex cluttered obstacles (see [1]).

Thus artificial and/or biological agents can develop sensitivity to a core set of important discriminative features, and dynamically change their perspective of the environment till the required regularities are highlighted.

The second approach using Deep Convolutional Neural Networks falls into the category of passive vision systems. It differs from the former Active Vision controller by using a much larger Neural Architecture with a more supervised training phase and only providing outputs which predict the road's location on the image plane. These outputs are used by a simple differential control loop to make the final steering decions. For view-invariant applications, such as detecting roads in 'one-shot' from a static camera input, deep convolutional networks with their stacked internal representation of the input space share many similarities with biological vision systems(see [66], [22]). Convolution networks are powerful due to their ability to transform small local image patches into higher level feature representations. Thus having such a model that carries information of a wide set of universal features that can be found in roads irrespective of the environment could result in more robust and general-purpose detection and compensate for the lack of dynamic active perception capabilities.

Before proceeding into further theoretical detail on these two approaches in Sections 2.3.1 and 2.3.2, the sensory and computational apparatus required for our proposed solutions needs to considered. While integrating camera inputs with 3-D laser data can have benefits (see Section 2.1) we chose to build purely vision based solutions for following. LIDAR scanners can be prohibitive in terms of cost and may not be readily available in some cases. Moreover, in theory roads by their very nature can essentially be distinguished from their surroundings in terms of differences in colour, if the algorithm which processes it is sensitive to it and the data presented to it is not corrupt. For example, even in the extreme case of desert-tracks, regions where tire-marks are laid out show subtle differences in color which can give enough information for navigation decisions. Another important factor is the on-board computational resources required to run the proposed road-following algorithms in real-time. This may not be as much of an issue in the current day as it was historically, with the recent advent of GPU computing. However if large-scale deployment is considered, computational resources still remain an important issue. A road-following solution that can

run on a basic computational hardware and is not dependent on an expensive sensor suite may be preferable to one with similar levels of performance but with higher computational and sensory requirements.

### 2.3.1 Evolutionary Embodied Active Vision

The theory of active perception is based on the assumption that any perceptual process depends as much on the sensory apparatus characterizing an organism as on its motor activity. Within an active approach to perception, active vision refers to the sequential and interactive process by which an agent "actively" selects and analyzes parts of the visual scenes [85]. The famous "Kitten in the Gondola" experiment illustrated in [49] clearly demonstrated that normal visual development depends not only on movement of the body relative to the environment, but also on self-actuated movement. In the last 30 years, a fast growing body of literature in psychology and neuro-science forwards the case of looking at action and perception as part of unified closed sensory-motor loop. In the domain of robotics, the active perception paradigm has been implemented following different approaches, in particular for vision-based tasks. For a comprehensive review of active vision in cognitive science and robotics we refer the reader to [25]. In this section, we focus mainly on a particular approach to active vision based on the use of artificial neural network controllers synthesized by evolutionary computation techniques.

Evolutionary Algorithms (Artificial Evolution) refer to a family of meta-heuristic techniques that aim to use the principles of biological evolution such as natural selection, chromosome exchange (cross over), mutation etc. to find a solution for global optimization problems. These techniques are often preferred over standard optimization approaches for problems with complex and non-linear objective functions. They have been successfully applied for a number of real-world domains; including the work presented in [76] where artificial evolution was used for designing a space satellite antenna (see [26] for more examples). Genetic Algorithm [41], a popular evolutionary search technique, is often used for training dynamic neural networks when the traditional supervised gradient descent approach is not applicable or preferred. This field of research, called Evolutionary

Robotics (see [86] for a detailed overview), espouses the case of investigating cognition and autonomous behaviour in the framework of a dynamical system that is embedded and in constant interaction with its environment rather than an input-output system based on traditional computational metaphor [20], [115]. Examples of succesful application of evolutionary robotics to develop neural controllers that display adaptive behavour that would otherwise be prohibitive to progam manually or train through traditional supervised methods, can be found in the experiments published in [12], [21] and [82]. This approach of generating adaptive neural networks can be said to mimic the learning process in natural organisms; where development of intelligent behavior occurs through iterative adjustments to responses generated from self-induced explorations of the environment.

[19] and [115] state that apart from the external environment, developing sensorimotor behaviour at par with biological agents requires an artificial neural network to account for the platform it is embodied in or in control of, during the learning process. They state the importance of treating such systems as embodied and situated, that is that they have full control over their bodies/platforms, and by extension their future perception, as this is intrinsically linked with the environment that they operate, and crucially, develop in. From a road-following perspective this approach is completely novel when compared to the state of the art methods, which consider perception of the road and motor commands as two seprate control modules which are be interfaced together to produce the final behavior. Instead as demonstrated by these works, a neurocontroller must be given a high degree of autonomy of movement on the road during its learning/training phase to develop truly adaptive road-following behaviour. The development of perception and locomotion thus cannot be considered separately and an unified sensorimotor controller for road-following needs to be explored.

Logistical constraints mean that in most cases neural networks cannot be trained on the actual platform and environments they are meant to be operated on. It is impractical for a neural network to be simply given control of a real-world vehicle and ‘evolved’ on actual roads. Therefore it is preferable to perform artificial evolution in a simulated environment, with the assumption that the learning is robust enough for later application real-world scenarios. In the Radical Envelope of Noise hypothesis work [52], the authors

demonstrate that through careful design of a virtual environment, the limitations of simulation can be overcome and it can be employed to produce solutions at least as good as those that can be achieved by evolving on real-world agents. This shows that employing evolutionary robotics, through simulation, should be an effective framework through which road-following can be tackled.

With approach the neuro-controller is thus an integrated action-perception system in which the camera images, largely reduced in resolution, contribute to form the controller input vector, which may be made of readings coming from other sensors. The full image resolution cannot be directly input into the network, as having a larger input vector means increase in the number of trainable network parameters. This in turn massively increases the search space for the genetic algorithm and as a result the computational resources required to find a solution. The sensory information is then propagated forward to the network output layer, which sets the speed of the robot wheels, and/or the pan and tilt orientation of the camera, and/or the level of the zoom. In this system, the agents actions contribute to altering its perception of the scene and thus in turn affects the visual inputs it receives. This is referred to as the closed sensory-motor loop. Various papers have investigated the potentialities of this approach in different vision-based tasks, showing that the integrated neuro-controllers that close the action-visual perception loop manage to overcome the limitations imposed by the use of low resolution images to allow autonomous robots to perform complex visual discrimination tasks.

One of the pioneering works in the use of artificial evolution to develop situated active robotic vision systems is [44]. In this study, a robot suspended from a gantry frame has stepper motors that allow translational movement in the $X$ and $Y$ directions, and a CCD camera pointing down at a mirror inclined at 45° to the vertical. The task of the robot is to distinguish a white isosceles triangle from a white rectangle fixed to one of the black gantry walls by navigating towards the triangle. The contribution of this study is in showing the significance of the movement of the robot in carrying out the discrimination task. The work described in [21] explores the possibility of evolving neuro-controllers for mobile robots that can use their visual perception to perform tasks which are difficult or impossible using only proximal sensing. The task chosen is to move to the center of a cylindrical arena, and

stay there. The authors show that relatively small non-modular neural network controllers that process low pixel resolution images, can compensate for this deficiency by generating the actions that bring forth the most informative sensory stimulation, which in turn is used to generate task-effective actions.

Experiments to solve a perceptual task (differentiating between a rectangle and triangle) were also carried out in [60], with a network having only nine cells as its visual input or “retina". The network however had additional feedback units, which gave it the ability to zoom in and out of the image plane. It was also able to control the filtering strategy used to reduce pixels into the final nine values that were fed into the input layer. Thus each input neuron corresponded to an area on the image plane, the size of which was determined by the network’s output neuron. A regular feed-forward neural network which remained static and considered the entire image plane in its input vector was also tried to solve the same task. The network was trained using back-propagation under different configurations (number of hidden layers, learning rate and momentum of the gradient descent algorithm). The results showed that under no circumstances could this “static” network succeed in the task of differentiating between triangular and rectangular shapes, despite its higher resolution visual receptive field. The robots were required to carry out vision based collision avoidance. The results of this work indicate that, as it is the case with biological agents (e.g., the “Kitten in the Gondol” experiment mentioned earlier), neural networks with active body movement out-performed their “passive” counterparts. The difference in development of neural networks receptive fields under active and passive vision conditions is further explored by the same lab (Laboratory of Intelligent Systems, Swiss Federal Institute of Technology (EPFL)) in [109] and [110]. In both cases, a recurrent neural network (with hidden units) was evolved to control a mobile robot required to travel collision-free in a walled arena. Networks were evolved for maintaining a straight trajectory during the trial. The network had a feedback unit similar to that in [60] controlling the filtering strategy (i.e., which pixels of the raw image are combined to form the final input vector). The network outputs are also used to set the pan and tilt camera orientation, as well as the speed of the robot wheels. In [109], it is shown that a network evolved in a simulated environment with active vision capabilities (and then ported to the robot) develops sensitivity to a different set of features

as compared to a network which is developed by training on static snapshot images of the operational environment. Indeed, as mentioned in [108], neural networks with active vision capabilities develop sensitivity to fewer but more important features in the environment and successfully carry out tasks by maintaining a fix on them. In [110] the importance of active body control for the development of visual systems in neural networks evolved to control mobile robots is further demonstrated. A neural network controller was evolved for the same navigation task described in [109], and two sets of experiments were carried out. In one set an online learning rule (Hebbian plasticity) was applied and the weights of the neural network controller were updated as the network moved in the environment. In the other trials, the weights were also updated, but the robot was only free to move its camera (through pan and tilt outputs), not its body (motor outputs were ignored). After a number of iterations the learning was stopped and the networks with restricted movements were free to move. It was observed that only networks that had retained active body movement during the learning updates could complete the task successfully, while those that were unable to move the robot were unable to carry out the navigation task and avoid colliding with the arena walls.

Besides active vision abilities, it is also important to consider the effect of neural networks which can integrate information over time (recurrent networks) as opposed to being completely reactive (regular feed-forward). This is investigated in [107], where a recurrent network (similar to the architecture described in [109] and [110]) was required to differentiate between two patterns and to navigate a mobile robot to a goal destination based on the pattern discrimination task. The two patterns were on opposing walls, while the network's visual receptive field was too small to distinguish one from the other instantly in one update cycle. The network was able to solve the task by scanning the walls, sequentially searching for features and integrating this information to identify larger patterns. Based on these principles, it was demonstrated in [34] that complex machine vision tasks such as automatic driving and indoor navigation could be solved by simple low-resolution artificial neural network controllers when implemented as integrated action-perception control systems wherein information from one update cycle is retained to influence subsequent outputs. The automatic driving task consisted of a network driving a car in different courses

in simulated mountain roads. Results indicated that the best evolved networks performed as well as or better than human drivers tested on this virtual platform. Similar to other experiments, this relatively simple network with active vision capabilities was able to successfully navigate these roads and even carry out sharp maneuvers at bends by focusing on and maintaining a fix on one simple feature (such as the far edge of the road).

Largely inspired by the above mentioned research works, the road-following solution presented in Chapter 3 embraces this embodied active vision approach. Artificial neural network controllers that are structurally conceived to be integrated action and adaptive color perception systems are synthesized through evolutionary computation techniques. Starting from the assumption that the robot's operating conditions feature a potentially detectable difference in color between the road and the non-road area, we aim to integrate into the robot's controller a perceptual apparatus capable of dynamically adjusting the way in which the robot senses colors in order to cope with the variability of the real world.

### 2.3.2 Deep Convolutional Neural Networks for Road-Detection

Over the past decade, usage of 'deep' neural networks network architectures (deep learning) has led to drastic improvements in the state-of-art for a number of computer vision, speech and natural language processing problems. They are increasingly being preferred in research and industry over traditional feature-engineering based approaches for their ability to form robust hierarchical representations of the input space; a trait mirrored in biological organisms [66], [22]. While this concept of using multiple hierarchical processing or hidden layers in neural networks to improve results was always understood from parallels in biology, their implementation has been made practical by the recent advent of GPU computing, ready availability of large datasets and modifications to traditional gradient descent learning (back-propagation) initially proposed in [73]. Convolutional neural networks (see [71]) for vision problems are formed of layers of filters (convolution matrices) whose parameters are represented in the form of trainable weights, often followed by regular feed-forward layers towards the end of the structure. After most convolution layers, a form of non-linear down-sampling referred to as pooling is performed by usually considering the

mean or maximum value (average of max-pooling) of localised receptive field responses.

Compared with more conventional design based computer vision techniques and traditional machine learning algorithms, deep convolution neural networks offer significantly higher accuracy in bench-mark computer vision datasets for handwriting recognition (MNIST [73]), general image classification (ImageNet [69]), facial recognition/verification (LFW [111]), integrated object localisation and detection (ILSVRC2013 [100]) etc. As mentioned in [72], convolutional networks take advantage of the property that many natural signals are compositional hierarchies in which higher level features are obtained by composing lower-level ones. The widely used AlexNet network ([69]) with five layers of stacked filters has been found to represent colour and gradient based features in the convolution kernels of its first layer; with progression towards more abstract higher level features in layers above (see [79]). In the work presented [15], it was shown that similarly hierarchical representation of visual information can be found in the IT (inferior temporal) cortex of primates. Biological analysis of cortical areas associated with visual tasks in primate brains shows that more complex and higher-order representations are also more robust to variations and the exact appreance of features they correspond to in the retinal field. [33]. This concept of being sensitive to regularities in the visual input, irrespective of geometrical transformations (scaling, rotation and translation) and illumination conditions is key to the success of convolutional network. It should be noted that the commonly used low-level feature descriptor, SIFT [78] was also inspired from this attribute of biological vision. The filtering and pooling layers, which are the core components of convolutional neural architectures are themselves inspired from the classical neurological concepts of simple and complex cells (see [50]). In this classic study the simple cells were shown to be neurological units corresponding to small localised fields within the wider receptive field, like convolution matrices, while the position-invariant complex cells cover wider areas, integrating responses from multiple simple cells like pooling/sub-sampling layers.

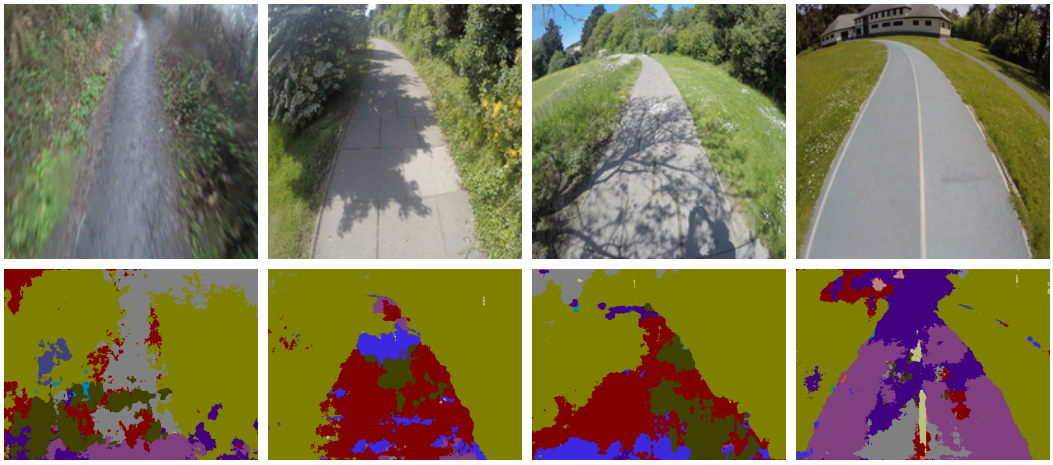
Besides ‘depth’ modern convolutional networks also benefit from having wider processing layers; incorporating a relatively high number of units (either filters or feed-forward neurons) in each layer when compared to artificial neural models of the past. The network can thus be sensitive to a wide array of regularities at each processing level. This ability

to compose varied abstract features, especially at the highest layers, from the initial raw-input/receptive field offers potential to problems such as road-following where it is difficult to design or train models that encapsulate a global and generalised representation of the environment. Thus with the growing usage of deeper neural architectures, researchers have been trying to apply deep convolution neural networks to various instances of the problem of detection and navigation on a road by autonomous vehicles.

Deep convolution networks are trained by variations of the standard gradient-descent approach and often require relatively large datasets to prevent the learning from being 'over-fitted'. Indeed the sizable structure of these networks containing millions of trainable parameters which offers them their unique advantage, can also lead to the learned features being too similar and 'specialised' in cases of limited and/or homoegenous training sets. However more unsupervised and biologically relatable learning methods similar to the evolutionary embodied approach described in Section 2.3.1 are not considered plausible due to the difficulty of optimization techniques such as genetic algorithms finding an appropriate solution in the vast search space. Researchers instead incorporate elements of unsupervised learning through a technique called 'pre-training' (see [53]) wherein a network already trained on a larger dataset is structurally modified at the final layers and subject to another round of training or 'weight fine-tuning' on images specific to the intended application (see [70] for an example of this approach).

A rare instance (across all domains) of the use of non gradient-based learning algorithms is the series of autonomous driving experiments described in [64] and [65]. The authors used the principles of evolutionary robotics to generate a convolutional network, serving as a high-level feature extractor to a small recurrent network controlling a virtual robot car in the TORCS [120] racing simulator. Initially the convolutional network was evolved [64] offline on pre-captured snapshots from TORCS test tracks using a simple fitness function which relied on maximizing euclidean distance between the feature vectors generated from each image. A small recurrent network outputting three driving commands (left turn, right turn, throttle/break) was then added on top of the final convolutional layer and itself evolved 'online' on the track, with total distance travelled as its fitness function. The same approach was followed in [65], except both the convolutional and final recurrent

network were co-evolved as part of a single structure. The former off-line approach exhibited marginally better performance when evaluated virtually on TORCS. However both approaches were out-performed by a standard recurrent network (presented in [63]), evolved under similar conditions and receiving the full-resolution ($63 \times 63$) of the input image as its input. While sequences of successful driving behaviour from such evolved convolutional networks was indeed demonstrated, the architecture used is quite limited in terms of the width of each convolutional layer when compared to the standard used for general object recognition in [69] as well as other deep-learning based road-following works. See Chapter 4.4 for further discussion on these works and results from an initial experiment in which we similarly evolved and tested a relatively small convolutional neural network in a set of simulated environments. The results of this experiment were however poor when compared to more simplistic neural models and led to the need to consider training convolutional neural networks using more traditional methods.



(a) (b)

Figure 2-1: The top row shows raw images of two road environments. The bottom row shows the corresponding pixel wise classification of these road images using the deep convolution neural network described in [10]. Purple pixels (almost completely absent in these images) refer to parts of the image which the network identifies as the road.

Most of the work using convolution networks for automatic driving has been focused and evaluated on highways and urban roads. An exception to this is the work described in [42], wherein a controller which was able to autonomously drive on highly delineated off-road terrains was developed. A convolution network (trained offline) was used as a fea-

ture extractor providing a robust representation of complex environments. Terrain in front of the robot was segmented into multiple categories by a classifier, which was trained on-line using self-supervised data labels. In [17] and [75], a deep convolution neural network was trained using hours of footage taken from a human driving a car in a racing simulator. Both implementations outperformed a baseline method where road detection was carried out using pre-defined features based on Gabor filters (see [90]). While the neuro-controllers were successfully tested on real world video sequences, these were limited to a subset of roads found in urban environments. Similarly a convolution network for lane detection on highway roads was trained in [51] using a large data set of video sequences collected over the course of multiple days. Using this approach over more traditional and simplistic 'line detector' methods meant that lanes could be accurately estimated even when there were occlusions and environmental disturbances. In [10], a de-convolution (convolutional encoder-decoder) network was used for semantic pixel-wise image labeling. The results show that the network is capable of successfully parsing and segmenting urban road scenes into multiple categories such as roads, pavements, trees, etc. The results also show that the network performs significantly better than various other machine learning methods. The method described in [4] also relied on scene segmentation by a convolution network detecting the road. Classification from the network is aggregated with that from a statistical color-information based method. With this approach the generalization capabilities of the convolution network which had learned high-level features from other road scenes could be combined with the color-based method which could adapt to dynamic changes in the current road. One of the key aspects of this work was that the convolution network was trained in a semi-supervised manner on road images using noisy labels generated by classifiers trained on a larger general image data set. A weakly supervised method of generating training labels can also be found in [11], where a convolutional network performing pixel level segmentation is trained and evaluated (off-line) on urban road datasets. However considering the review of similar techniques in Section 2.1 the use of 3-D laser data and pose estimation techniques based on vehicle and geometrical priors to detect the road during the label generating phase may not be appropriate for more challenging roads.

To the best of our knowledge, convolution neural networks such as those described

in [10], [17] and [4] have not been evaluated on non-urban roads, and their performance in such environments is yet to be ascertained. We have tested the system described in [10] on various images of poorly delineated roads, some of which taken from the environments where we conducted the experiments described in this study. The performance of the system turned out to be relatively poor (see Figure 2-1). The multi-modal feature fusion approach described in [117] espouses the use of deep CNNs trained with features from RGB, Depth and Near Infrared Data to perform semantic segmentation of road scenes (similar to [10]). Whilst promising with the use of multiple varied input sources, further evaluation is required in more varied and complex conditions (as mentioned by its authors). These networks trained specifically for off-terrain, forest tracks may not be able to generalise to urban and high-way road scenes. The work carried out by NVIDIA described in [13] is one of the best performing examples of the application of deep CNN models for road-following till date. Inspired by earlier efforts to use simpler feed-forward architectures to directly control a vehicles wheel speeds (see [94]), the authors propose inputting images encoded with the YUV colour model to deep CNNS that directly output steering commands. The authors train the system with labels generated from sequences of manual driving. The dataset is standardised and augmented to account for a greater portion of images corresponding to turns whilst discarding sequences that are very similar to each other. Although the trained CNN has been shown to generalise and control an autonomous vehicle in challenging conditions such as unpaved roads with fuzzy boundaries, the authors state that the system needs further validation and increased robustness. Indeed while the vehicle can drive fully autonomously for large sequences, there are still instances in which manual intervention is required to remain on-course.

There is therefore further scope to explore road-following in undefined and complex environments using these powerful neural structures which can form given well constructed training/evaluation datasets and robust representations of colour. It is acknowledged that the same issue of biases generated from human selected training data (associated with other neural network based works in Section 2.2) also exists with the use of convolutional networks trained through back-propagation. Moreover the concepts of recurrent nodes, embodiment, action-generated learning etc. are totally discarded this approach. However

despite being an essentially static passive vision road-detection solution, the robust view invariant representation of features at multiple levels, with similar traits observed in studies of mammalian vision, makes deep convolutional neural networks an approach worth exploring.

# Chapter 3

# A dynamic active-vision controller

The road-following solution proposed in this chapter is largely based on a particular implementation of the "active vision" approach in robotics based on the use of artificial neural networks synthesized by evolutionary computation techniques discussed in the previous chapter (Section 2.3.2). This approach, described and tested for visual discrimination problems in various papers (e.g. [44], [109]) is extended to color vision for developing a neuro-controller with dynamic color perception capability that successfully navigates a mobile robot in various unmarked roads. The "active" aspect of our controller mainly resides in a unified closed sensory-motor loop in which sensation and in particular color sensation is determined by its own action. The distinctive contribution of the experiments presented in this chapter is to develop and test the effectiveness of artificial neural networks in generating an integrated action and adaptive color perception system that can drive a robot on various roads which differ in the combination of colors of the road and non-road surfaces.

Initial experiments involved evaluating evolved neurocontrollers in variations of the to determine the composition of virtual road scenarios best suited for the emergence of dynamic colour perception (see Section 3.2). After this exploration of evolutionary conditions, an evolved neuro-controller was ported onto a Pioneer-3AT robot for road-following trials in a set of road conditions created in an indoor labaoratory. Results from this initial set of indoor trials (presented in Section 3.3) indicated that neural networks could successfully control a mobile robot in roads, with conditions much different (in terms of visual composition and colour distribution etc.) from those that formed the basis for its training. At this

stage it was hypothesized that, keeping the same configuration of evolutionary conditions, further performance improvements can be achieved by considering alternate models of representing colour. Indeed as discussed in the previous chapter, the *RGB* colour model used for real and virtual trials thus far can be ill-suited for the greater degree of environmental noise in outdoor conditions. To investigate representations of colour that are best suited for outdoor operation, we conducted a set of simulated trials wherein 504 different permutations of colour channels from six models are explored. After evaluating the network's performance using different combinations of color models in a vast range of virtual environments (Section 3.4), we demonstrate that the network can reliably navigate a real robot in real world roads (Section 3.5). A functional analysis of the neuro-controller's mechanisms for dynamic colour perception in Section 3.6 sheds light on complex but effective strategies generated during the evolutionary learning phase in simulated road environments which enable it to capture structures and regularities while driving in challenging real-world conditions using extremely low pixel resolution images as input.

## 3.1 Methodology

The premise behind the road-following approach described in this chapter is to use a continuous time recurrent neural network or CTRNN (see Section 3.1.2 for details) that controls the wheel speeds of a mobile robot platform (Pioneer 3-AT). This neural architecture has only 25 input nodes, 6 hidden recurrent nodes and 7 output nodes. 3 of these output nodes control the ratios with which the individual colour channels of the raw input image are mixed down to create the final input vector. The network is trained using the principles of embodiment and evolutionary robotics discussed in section 2.3.1. The training scheme involves a population of networks controlling a virtual robot having the same kinematic properties as the Pioneer 3-AT on a set of simulated roads, rendered with textures designed with specific colour composition properties to encourage the evolution of solutions that have the mechanisms to adapt to more complex real-world scenarios. The colour properties of these virtual roads are designed such that the network is forced to dynamically vary each of the 3 colour feedback nodes to achieve maximum fitness i.e. navigate all the training

road scenes.

Thus the use of a simulated environment serves two purposes. Firstly the resources available to us meant online evolution on the target mobile platform was not feasible. Besides through careful design of the properties of the simulated environment, we could add selective pressures that encourage the emergence of adaptive solutions. The colour model used during training was RGB although as shown in Sections 3.4 and Sections 3.5, these networks perform better when alternate colour representations are used. It is acknowledged that the use of alternate colour representations during training could have an effect on the performance of such evolved networks. However it was assumed that since the network essentially receives a stream of real valued numbers as its input the use of any particular colour scheme during training should not lead to fundamentally different behaviour as long as the same technique to encourage the use of all the colour feedback nodes is employed. Subsequent subsections discuss individual aspects of the neuro-controlelr architecture and training methodology in more detail.

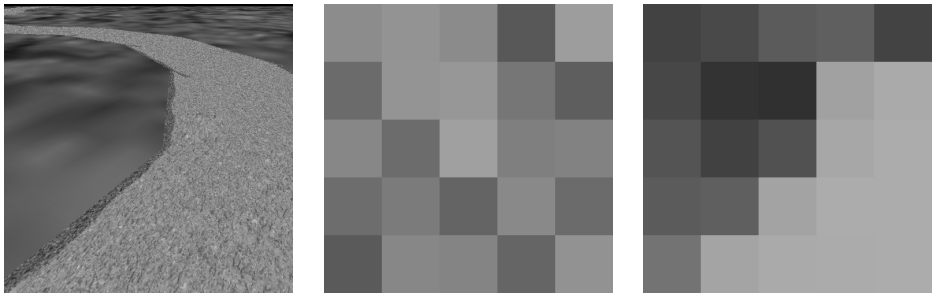
### 3.1.1 Visual Receptive Field

The proposed neuro-controller manoeuvres the robot by setting the speed of its right and left wheels, achieving differential drive, and at the same time generates three color parameters ($\rho$,$\gamma$, and $\beta$) which determine how much each colour component (e.g., $R$, $G$, and $B$) from the raw camera images contributes in generating the network input vector. The raw input image resolution of $500 \times 500$ pixels is significantly reduced by overlaying a $5 \times 5$ grid on the image. Each of the 25 grid cells cover an area of 10,000 pixels. This dimensionality reduction step is carried out to reduce the neural networks input field resolution to a reasonable size of 25 nodes, similar to the architectures used for mobile navigation and simulated autonomous driving in [109] and [34] respectively. Additional input nodes mean increased number of trainable network parameters which leads to a more complex and higher dimmensional solution search space for the evolutionary optimization algorithm. Pixel values from the 3 colour channels from each grid are averaged to compress and collectively represent the $10000 \times 3$ pixels as 3 real valued numbers. An alternative approach

to this resolution reduction procedure can be to randomly sample pixels from each grid; altough results from [109] where the network is given freedom to choose between either of these techniques suggest grid-averaging being more preferable. Representing a grid as the average pixel intensity of each constituent colour channel also means the network does not have to be learn to regularities in local colour distributions, but rather differences in contrast between 'higher-level' regions in the raw input field. Thus for each grid or cell, the normalized mean value of each RGB color component ($C_R$, $C_G$, and $C_B$) is computed by first summing the respective color component of each pixel within the cell and then dividing by the number of pixels in the cell. Each cell $i$ generates a sensory input $I_i \in [0,1]$ by combining its mean color components as follows:

$$I_i = \rho \times C_R + \gamma \times C_G + \beta \times C_B. \tag{3.1}$$

The parameters $\rho$, $\gamma$, and $\beta$ are real numbers in $[0,1]$ generated by the controller at each time-step (see Section 3.1.2). Their value is such that $\rho + \gamma + \beta = 1$ to represent the ratios in which the mean color components are mixed in forming the input of each grid cell. By varying $\rho$, $\gamma$, and $\beta$ the controller can dynamically adapt its perceptual system to the color characteristics of the environment. The controller can thus learn adaptive strategies to activate the neurons associated with these parameters; enabling it to extract discernible cues from the visual field in a manner that is robust to the colour properties of the road/non-road areas as well as environmental variations as the robot travels along a road (e.g. illumination, appearance of new objects etc). A permanently static combination of colour channels may not be enough to provide the neuro-controller with the perceptual information necessary to navigate roads across the entire range of possible operational scenarios. Results from bench-marking tests (presented in Section 3.4 of the following chapter) provide further evidence to support the use of controllers with this dynamic colour mixing attribute over similarly sized architectures that have no active vision capability with a fixed ratio of colour channels constitute their final input vector. See 3-1 for a visualization of the perceptual information available to the controller when a raw camera image is pre-processed and represented as a vector of 25 real valued numbers.



(a) (b) (c)

Figure 3-1: Formation of the neural network controllers final input receptive field from an initial raw camera snapshot of the road. (a) Full resolution ($500 \times 500$ pixels) monochrome snapshot of a simulated road in greyscale RGB. (b) View of the raw image/snapshot in (a) as seen by the robot controller after the grid-averaging dimmensionality reduction step described in Section 3.1.1 is performed and all three channels ($R$, $G$, and $B$) are mixed in equal proportions. Grids or cells that belong to the road cannot be identified from this visual perspective. (c) View of the raw image/snapshot in (a) as seen by the robot controller after the grid-averaging dimmensionality reduction step described in Section 3.1.1 is performed. Now the red and blue channels are completely discarded and only the green colour channel is highlighted (i.e. $\rho = 0$, $\gamma = 1$, and $\beta = 0$). The group of brightly colored pixels on the right hand side of the image representing the road can now be clearly seen after the perspective of the same visual scene is altered.

One of the key challenges for the evolutionary robotics line of research lies in formulating the evolutionary conditions that can lead to the emergence of desired behavior, i.e. ability to autonomously drive mobile platforms on roads of any type with adaptability for dynamic high-noise environments. As on-line evolution in real world scenarios is not practical, a driving simulator with a set of carefully designed road scenes was developed for training our road-following neurocontroller. Controllers are evolved and evaluated 'offline' in this virtual environment before being ported onto a mobile robot for autonomous driving trials. It should be noted that the evolutionary search process which generates the weights of the road-following neurocontroller does not converge on 'solutions' that can drive robustly in actual roads (as is desired by us); but instead provides solutions that improve performance in the evolutionary environment. Therefore by careful design of the environment we can hope that the strategies that help a successful controller navigate these virtual roads will also hold for the more complex case of real roads. As discussed in Section 2.3.3 of the previous chapter, the development of visuo-motor control in the case of both natural and artificial agents is linked to the brain or artificial neural-networks interaction with the

environment (real or simulated) through the body or robotic platform. See Section 3.1.4 for details of this virtual environment and Section 3.2 for discussion on the design choices behind the geometrical and colour properties of these simulated road scenes.

### 3.1.2 Controller and the Evolutionary Algorithm

The robot is controlled by a continuous time recurrent neural network (CTRNN) of 25 visual input neurons, 6 inter-neurons, and 7 output neurons. CTRNNs are a particular type of dynamic neural networks that have been extensively used in recent years as control systems for autonomous robots. Originally proposed in [12] as an alternative neuro-controller to classic discrete time artificial neural networks, CTRNNs are universal approximators of dynamical systems [36]. The main characteristics of this network is to provide the neural plasticity required to allow the robot to adjust its behaviour to the current operating conditions [43]. Memory and learning mechanisms are represented by the recurrent connections, as in classic artificial neural networks, and by the state of the network nodes which can vary overtime in response to the robot sensory experience [115]. See also [45] for an extensive discussion of the characteristics of CTRNNs as neuro-controller for autonomous agents. The structure of the network is shown in Figure 3-2. The states of the output neurons are used to control the speed of the left and right wheels as explained later, and to define the parameters $\rho$, $\gamma$ and $\beta$ mentioned above.

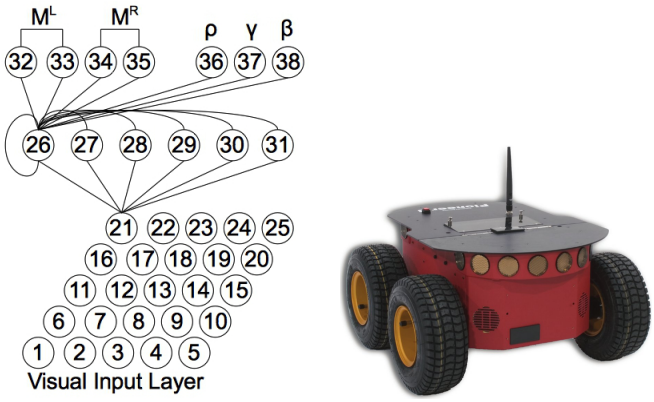
The values of sensory, internal, and output neurons are updated using Equations (3.2), (3.3), and (3.4).

$$y_i = gI_i \text{ for } i \in \{1,\cdots,25\}, \tag{3.2}$$

$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{31} \omega_{ji}\sigma(y_j+\mu_j) \text{ for } i = \{26,\cdots,31\}, \tag{3.3}$$

$$y_i = \sum_{j=12}^{15} \omega_{ji}\sigma(y_j+\mu_j) \text{ for } i = \{32,\cdots,38\}, \tag{3.4}$$

with $\sigma(x) = (1+e^{-x})^{-1}$. In these equations, using terms derived from an analogy with real neurons, $y_i$ represents the cell potential, $\tau_i$ the decay constant, $g$ is a gain factor, $I_i$ with




(a) (b)

Figure 3-2: (a) The neural network. The lines indicate the efferent (outward) connections for only one neuron of each layer. Each hidden neuron receives an afferent connection from each input neuron and from each hidden neuron, including a self-connection. Each output neuron receives an afferent connection from each hidden neuron. It should be noted that the $\rho$, $\gamma$ and $\beta$ parameters do not directly receive the values from the output nodes they are associated with but rather derived from the raw activations (see equation 3.1). (b) The Pioneer 3-AT robot.

$i \in \{1, \cdots, 25\}$ is the activation of the $i^{th}$ sensor neuron, $\omega_{ji}$ the strength of the synaptic connection from neuron $j$ to neuron $i$, $\mu_j$ the bias term, and $\sigma(y_j + \mu_j)$ the firing rate (hereafter, $\eta_i$). All sensory neurons share the same bias ($\mu^I$), and the same holds for all motor neurons ($\mu^O$). $\tau_i$ and $\mu_i$ with $i = \{26, \cdots, 31\}$, $\mu^I$, $\mu^O$, all the network connection weights $\omega_{ij}$, and $g$ are genetically specified network parameters. At each time step, the output of the left motor is $M^L = \eta_{33} - \eta_{32}$, and the right motor is $M^R = \eta_{35} - \eta_{34}$, with $M_L, M_R \in [-1, 1]$. For the color parameter,

$$\rho = \frac{\eta_{36}}{\eta_{36} + \eta_{37} + \eta_{38}}, \tag{3.5}$$

$$\gamma = \frac{\eta_{37}}{\eta_{36} + \eta_{37} + \eta_{38}}, \tag{3.6}$$

$$\beta = \frac{\eta_{38}}{\eta_{36} + \eta_{37} + \eta_{38}}. \tag{3.7}$$

Cell potentials are set to 0 when the network is initialized or reset, and Equation (3.3) is integrated using the forward Euler method. The integration time step is $\Delta T = 0.38$, as it takes the intended hardware platform, the Pioneer robots embedded computer 0.38 seconds to perform a complete network update cycle.

An evolutionary algorithm using linear ranking is employed to set the parameters of the networks [41]. We consider populations composed of $M = 100$ genotypes coding for the parameters of the robot controllers. At generation 0, each genotype comprising 243 real values (228 connections, 6 decay constants, 8 bias terms, and a gain factor) is chosen uniformly random from the range $[0, 1]$. Generations following the first one are produced by a combination of selection with elitism, recombination, and mutation. For each new generation, the highest scoring genotype ("the elite") from the previous generation is retained unchanged. Each of the other $M - 1$ new genotypes are generated by fitness-proportional selection from the 30 best genotypes of the old population. Each new genotype has a 0.3 probability of being created by combining the genetic material of two genotypes of the old population. During recombination, one crossover point is selected. Mutation entails that a random Gaussian offset is applied to each real-valued vector component encoded in the genotype, with a probability of 0.04. The mean of the Gaussian is 0, and its standard deviation is 0.1. During evolution, all vector component values are constrained to remain within the range $[0, 1]$.

During evolution, each robot is evaluated on 24 trials. Each trial is a sequence of 500 time steps (190 seconds) in each of which first the wheel speeds are computed by calculating the outputs of the network and subsequently position and orientation of the robot are updated. During evaluation, each robot experiences 12 different evolutionary scenes, which differ in term of color characteristics of the road and non-road surfaces (see Section 3.1.4 and 3-4 for more details). The evolutionary training phase is deployed the HPC-Wales computing cluster and executed on in 'parallel' on multiple cores in parallel using the Open-MPI (Open Message Passing Index [37]) framework. This is due to the unreasonably high computational cost associated with serially processing evaluation trials of generations of network populations on a single machine. Each evolutionary experiment involves a 100 processor cores distributed over multiple nodes, with every core running a parallel process to evaluate a member of the population of neural network controllers. After each round of evolutionary trials, fitness scores from the entire population of processes is collected by a designated 'master' process/core. The next generation of genotypes encoding weights of these virtual neuro-controllers is generated and then distributed for evaluation from this

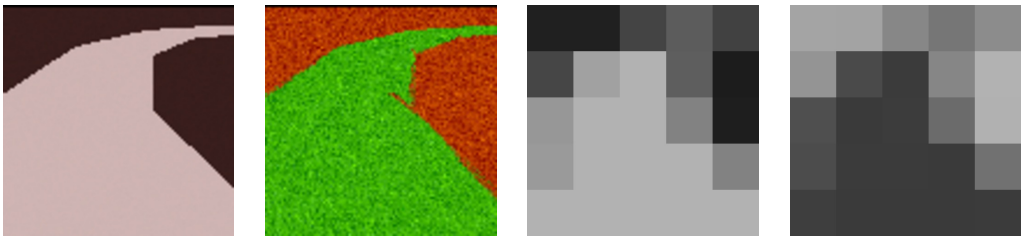
core. Simulated road scenes for the evaluations are rendered 'off-screen' using OSMESA, a popular open-source implementation of the OpenGL (`http://www.opengl.org`) graphics API.

### 3.1.3 Robot Platform and Kinematics

In the simulation, the robot is modeled as a circular object (radius 25 cm) with left and right motors which can be independently driven forward or backward, allowing the robot to turn fully in any direction. The motion control which is used to update the robots position in the virtual environment is based on the 2D two-wheeled differential drive kinematics model for mobile robots detailed in [32]. This model takes into account the robots structural parameters i.e. radius, wheel distance and speed-limits to give an output in terms of the robots updated position and orientation. The output of neuron 32 to 35 (Figure 3-2) are used to set the left and the right wheel speeds. The simulated robots maximum speed is set at 0.8 cm/s. Complex dynamical properties such as friction are not accounted for in this model. The authors of [6] highlight one of the several examples of the successful portability of this model from simulated to real-world platforms in the domain of evolutionary robotics. During the evolutionary training phase a random noise probability is modeled into the position outcomes of the virtual robot in an attempt to account for control scenarios on actual roads where friction, uneven terrain, motor power etc. may impact the robots motion.

### 3.1.4 Evolutionary Enviornment

During an evolutionary run, neuro-controllers are evaluated in 12 simulated road scenes which differ from one another with respect to the colour distribution of the textures used to render the road and non-road surfaces. Each of the 12 scenes is specially designed to encourage the neuro-controllers to dynamically vary all three colour feedback nodes. Additional scenes could have been added but this would result in an increase in training time and therefore the number of scenes was limited to only those that were necessary to encourage the emergence of adaptive/dynamic colour perception. A virtual camera, positioned on the front-end of the virtual robots body, looks at the 3D scenes rendered off-screen using



(1.a) (1.b) (2.a) (2.b)

Figure 3-3: Views from the 'virtual robot' looking at the road in simulation scenes and the corresponding input vector after the dynamic colour feedback has been applied to the raw image. 1.a and 2.a correspond to scene 2 and 1.b and 2.b corresponds to scene 7 (see table 3.1).

an implementation of OpenGL. The field of view of the simulated camera is set in order to match the characteristics of the camera mounted on the real robot. The camera positioning attempts to eliminate view of the sky, as it is assumed that part of the visual scene does not contain information necessary for road-following. However it is acknowledged that in highly inclined roads this assumption could fail. Each scene contains a tiled textured horizontal plane that represents the ground, and a texture deviated surface rendered on the ground that represents the road. At each simulation time-step, a $500 \times 500$ pixels image is captured from the virtual camera (see 3-3). These scenes have been devised in order to provide selective pressures to guide evolution towards the emergence of controllers that choose—by appropriately setting $\rho$, $\gamma$ and $\beta$ for each scene—the color components which assist the robot in distinguishing the road from the non-road surface, and disregard those that do not show the pattern that is being sought. With this set of scenes, the only means by which the robot can detect and successfully navigate all the roads is by varying, between trials, the color parameters $\rho$, $\gamma$ and $\beta$.

Snapshots from each of the 12 scenes are presented in 3-4. Refer to table 3.1 for the distribution of intensities in each colour channel ($R$, $G$, and $B$) for the textures used to render the road and non-road areas in these scenes. Scenes 1 to 6 feature only one color component. In these scenes, either the road is darker than the non-road surface, or vice-versa. For two other color components, a uniformly distributed random value in the interval $[0, 255]$ is assigned to each pixel of the scene. Scenes 7 to 12 feature two color components, one which is used to represent the road, and the other to represent the non-road surface. Similar

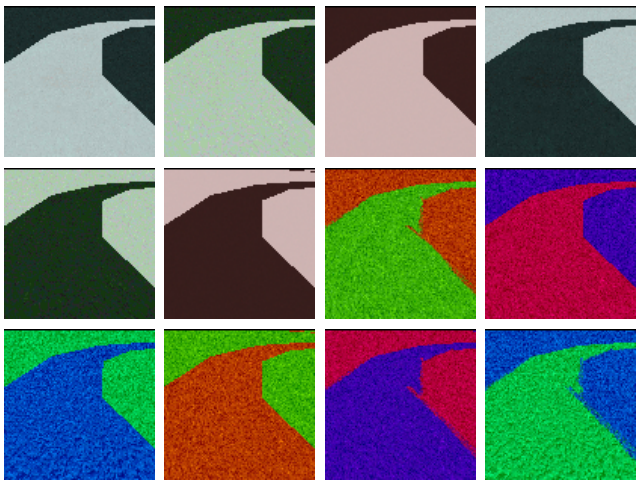


Figure 3-4: Snapshots of the 12 virtual evolution environments. The colour distribution properties of the textures used to render these are described in table 3.1.

to scenes 1-6, a uniformly distributed random value in the interval $[0, 255]$ is assigned to each pixel of the scene for the third colour component. In this way, the color component/s that are random do not provide any information that helps in discriminating between the road and the non-road surface. If greyscale RGB colour representation were to be used to compose the neural-networks final input field, there would be little or no contrast between grids belonging to the road/non-road regions. Indeed evolutionary runs conducted in these set of scenes with neural-architectures without the dynamic colour feed-back feature (using greyscale RGB) do not show any meaningful increase in individual and collective fitness. Thus the colour properties of the road/non-road textures for these scenes are designed such that to extract perceptual information required for navigation, in scenes 1-2 the red channel needs to be highlighted while simultaneously dampening the contributions from green and blue, scenes 3-4 require green to be highlighted with red and blue being the noise channels and scenes 5-6 similarly require the blue channel i.e. the *beta* parameter needs to be activated above those corresponding to the red and green. While scenes 1-6 can be considered sufficient for the emergence of dynamic variation of all three colour parameters, scenes 6-12 are included to provide further variation of scenarios that require an active approach to colour vision. In this second set of scenes the information required for road-following can be extracted by varying any 2 of the 3 colour parameters. Further detail on the design and colour composition of the evolutionary environment is provided in Section 3.2.

The evaluation scenes have average inherent contrasts between the road of non-road intensities of either 150 (scenes 1–6) or 120 (scenes 6–12) on a scale of 0 to 255. Preliminary tests showed that in scenes with a contrast below 80 and in lower contrast textures no successful controllers could be obtained (see Section 3.2). Real world environments can have contrasts much lower than this. Thus, the visual input on the real robots has been generated by applying histogram equalization to $C_R$, $C_G$ and $C_B$ in order to increase the global contrast.

Table 3.1: R,G,B colour distribution of textures used in creating road scenes. 'R' indicates uniform random distribution in 0–255.

| Scene | Road | | | | | | Non-Road | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Red | | Green | | Blue | | Red | | Green | | Blue | |
| | Avg | Sd | Avg | Sd | Avg | Sd | Avg | Sd | Avg | Sd | Avg | Sd |
| 1 | 57.4 | 1.5 | R | R | R | R | 207.4 | 1.6 | R | R | R | R |
| 2 | 207.4 | 1.6 | R | R | R | R | 57.4 | 1.5 | R | R | R | R |
| 3 | R | R | 29.0 | 6.3 | R | R | R | R | 179.0 | 6.3 | R | R |
| 4 | R | R | 179.0 | 6.3 | R | R | R | R | 29.0 | 6.3 | R | R |
| 5 | R | R | R | R | 47.2 | 4.1 | R | R | R | R | 197.2 | 4.1 |
| 6 | R | R | R | R | 197.2 | 4.1 | R | R | R | R | 47.2 | 4.1 |
| 7 | 59.9 | 20.8 | 173.4 | 21.5 | R | R | 178.5 | 20.4 | 53.4 | 21.3 | 10 | 10 |
| 8 | 178.5 | 20.4 | 53.4 | 21.3 | R | R | 59.9 | 20.8 | 173.4 | 21.5 | R | R |
| 9 | 56.4 | 21.9 | R | R | 168.2 | 21.7 | 178.8 | 21.9 | R | R | 59.6 | 20.0 |
| 10 | 178.8 | 21.9 | R | R | 59.6 | 20.0 | 56.4 | 21.9 | R | R | 168.2 | 21.7 |
| 11 | R | R | 185.9 | 25.7 | 65.3 | 25.0 | R | R | 66.3 | 26.1 | 185.9 | 24.8 |
| 12 | R | R | 66.3 | 26.1 | 185.9 | 24.8 | R | R | 185.9 | 25.7 | 65.3 | 25.0 |

For generating the trial course in each of the 24 evaluations, a total of 11 road tiles are used each 160 cm long and 100 cm wide. The length of the road the robot needs to travel is 1760 cm. Each road starts off with a couple of straight tiles followed by a smooth 30° bend left or right. This is followed by a similar smooth bend, with probability $\frac{6}{7}$ of it being in the opposite direction as the first one. This provision allows a robot to demonstrate the ability to make both kinds of turns and ensures the robot needs to be constantly maintaining its course to stay on the road. Subsequent turns are random, but checks are made to ensure no self-intersecting road shapes and $180^o$ 'U-bends' are generated. Moreover the maximum curvature between two successive road tiles is limited to $35^o$. During the 24 trials, each evolutionary scene is presented twice, first with a right turn followed by a left turn road,

then with a left turn followed by a right turn road. In order to simulate roads with amorphous nondescript edges, the edges of a set of road textures were manually faded out using noisy paintbrush tools in image manipulation software, and then alpha-blended with the underlying ground texture. However the use of such blended textures provides no observable difference to the properties of the evolved networks. This can be attributed to the grid averaging strategy that greatly reduces the resolution of the raw input, such that the texture edges do not make a large enough contribution to the intensity levels of grids in the final visual receptive field.

### 3.1.5 Fitness Function

In each trial $e$, the robot controllers fitness $f_e \in [1.0, 1.5]$ corresponds to the number of road tiles traversed from trial start, and the position in the last traversed tile, in case the robot does not reach the road end within 500 time steps. A trial is terminated earlier if the robot is detected to have moved off the road. The final fitness $F$ is computed as:

$$F = \Big(\frac{1}{E}\prod_{e=1}^{E} f_e\Big)^E \times \Big(\frac{1}{E}\sum_{e=1}^{e=E} C_e\Big), \tag{3.8}$$

$$f_e = 1.0 + \frac{K+H}{22}, \quad H = \frac{Q-V}{Q}, \tag{3.9}$$

$$C_e = \begin{cases} \frac{1}{S}\sum_{s=50}^{s=S} |R_s - P_s^1| + |R_s - P_s^2| & 1 \le e \le 6, s > 50, \\ \frac{1}{S}\sum_{s=50}^{s=S} 2 \times P_s^3 & 7 \le e \le 12,\ s > 50, \\ 0 & \forall e, \text{if } s \le 50, \end{cases} \tag{3.10}$$

where $E = 24$ is the total number of trials, $K$ the number of tiles crossed, $Q$ the tile length, $V$ the error vector from mid-point of the tile side closer to the road end to the robot position at the end of the trial, $C_e$ the quality of the dynamic color perception strategy in trial $e$, $R_s$ the value of the color parameters (i.e., either $\rho$, $\gamma$ or $\beta$) that has to be used to discriminate between road and non-road, $P_s^1$ and $P_s^2$ the values of the color parameters (i.e., $\rho$, $\gamma$ or $\beta$) that do not discriminate between road and non-road in mono-color scenes (i.e., scenes 1 to

6, Table 3.1), $P_s^3$ the value of the color parameters (i.e., either $\rho$, $\gamma$ or $\beta$) that does not discriminate between road and non-road in dual-color scenes (i.e., scenes 7 to 12, Table 3.1), and $S$ the number of time steps completed by a robot in each single trial $e$. $C_e$ is computed only after time step 50, to allow the robot some time to adjust its color parameters in an optimal way given the characteristics of the current scene. Robots leaving the road before the $50^{th}$ time step get a 0 fitness for that trial irrespective of the distance traveled since trial start. $H$ is set to zero if the robot crosses the road/non-road border. The term $\eta$ is introduced to guide evolution towards the emergence of solutions which vary in an adaptive way the activation of the color parameters. We observed that without the term $\eta$ the fitness function is not capable to steer evolution towards the emergence of the adaptive mechanisms required to visually discriminate the roads in all the evolutionary scenes.

## 3.2 Designing the Evolutionary Environment

Initial experiments involving the evolutionary active vision approach suggested that the emergence of adaptive dynamic colour perception cannot be be achieved by simply incorporating nodes associated with the $\rho$, $\gamma$, and $\beta$ feed-back parameters in the neural architecture. When evaluation scenes are rendered using textures aiming to represent real-world conditions (such as grass, dirt, asphalt etc.), there may be higher inherent contrast in some colour channels over others. Such evolutionary conditions do not lead to neuro-controllers that can dynamically alter the contributions of all three colour channels, as degrees of contrast however marginal can be extracted with a static combination of $\rho$, $\gamma$, and $\beta$. While such controllers can demonstrate successful autonomous driving in the scenes similar to those they were evolved in, their performance comes into question when assessed in more complex virtual and real-world scenarios. As a result of this road textures were devised wherein certain colour channels were made totally non-contributory for road-following by replacing their actual pixel intensities with a random noise signal.

Typical evolutionary experiments involve 24 evaluation trials (12 scenes presented with 2 road shapes each) and running for a maximum of 4500 generations last anywhere between 24-36 hours (when implemented using parallel-programming the HPC-Wales com-

puting cluster). For a specific configuration of the evolutionary environment multiple differently seeded simulations are executed, each seed corresponding to a particular sequence of random numbers that determine the synaptic weights of the initial population of network controllers. Ideally we would want to asses each generation of evolved neuro-controllers on a larger number of evaluation trials. However this leads to increases in the run-time and number of computation cores required for executing evolutionary runs that are beyond the capacity of the computational resources allocated to us in the HPC-Wales cluster. Thus rather than expose the controllers to a large number of evolutionary trials to account for real-world environmental variability; we focus on creating a limited set of scenes that encourage the development of adaptive behaviour that can be applicable for road-following in the intended operational conditions.

As mentioned previously in 3.1.4 the 12 scenes have either one (scenes 1-6) or two (scenes 7-12) non-contributing colour channel. To determine the inherent contrast levels for the non-random channels that carry visual information, the scenes are created in three formats which differ in terms of the intensity difference between the dark and the bright colours (see Table 3.2). 10 differently seeded evolutionary runs were carried out for each of these 3 conditions. There was also a need to explore the pros/cons of employing a smaller subset of these road-scenes (instead of all 12) which can reduce computational time whilst developing 'active perception' for all three colour channels. With this objective a further 10 evolutionary runs or simulations were executed with six evaluation scenes, three mono-colour, i.e. only one information containing channel (scenes 2, 4, 6) and three dual-colour i.e. two information containing channels (scenes 8, 9, 10), with inherent contrasts of textures from set *A*, see table 3.2).

Due to the nature of genetic algorithms and the complexity of the problem, not all experimental runs were able to evolve a successful solution. Only those experimental runs with fitness values high enough to indicate the ability to solve more than half of the evaluation scenes were selected for post-evolution tests described subsequently in this section.

Table 3.2: Contrast and colour distribution characteristics for the three sets of scenes.

| Set | Contrast between mean intensities of road and non-road (0 - 255) | Range of distribution of intensities (0 - 255) |
|---|---|---|
| A | 120 for all scenes | 120 for all scenes |
| B | 150 for mono-colour, 120 for dual-colour | 10 for mono-colour, 30 for dual-colour |
| C | 80 for all scenes | 80 for all scenes |

**Test 1**

In this first round of post-evolution assesment, controllers with the highest fitness from the last 500 generations of eleven successful evolutionary runs, refered to as 'solutions' for the remainder of this section, are subject to a uniform set of eight road shapes in each of the twelve scenes; with each road generated to be approximately 24 meters long. For each individual neurocontroller the inherent contrast levels in the scenes are kept the same as those experienced during evolution. The response of controllers to previously unseen lower contrast levels is discussed later in Section 3.2. The road shapes consist of two basic types, an "S" shaped course where the robot needs to make turns in both directions to reach the end and the other where a constant turn in one direction is followed by a straightening of the path. Each of these is generated twice with initial left and right turns for two different angles (20° and 30°) which dictates the curvature of these turns. During evolution the angle of curvature was always 30° and the road generation algorithm ensured that the overwhelming majority (6 out of 7) of shapes generated would be of the first "S" shaped type. The rationale behind generating this fixed set of road shapes was to discover the actual best performing individuals in the population. It is possible that the high fitness values of some controllers could be the outcome of chance rather ability to navigate multiple road shapes across all the environments. Data from re-evaluation tests also provides insights into the performance of the chosen controllers across each of the twelve scenes and as a result the effectiveness and flexibility of the evolved dynamic colour perception strategies.

A normalized distance score ranging from 0 to 10 is used to assess performance in each testing condition. Individuals that managed to reach the very end of the road in a particular scene would thus get the highest possible score of 10. Figure 3-5 shows the average of this normalized distance score in each of the twelve scenes. Only data for solutions of

evolutionary runs that used six scenes is included here. Figure 3-6 shows the same, but for solutions when twelve scenes were used during evolution. As during the evolutionary stage, the number of time steps (iterations) in each trial is fixed at 500. Thus individuals with higher scores not only demonstrated better strategies to stay within the road-boundaries but also greater speeds as they moved along the course.

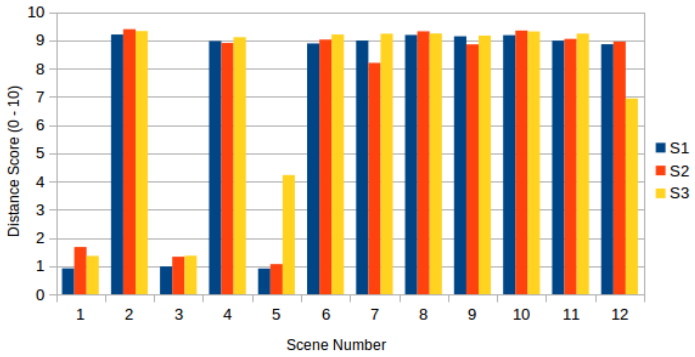



Figure 3-5: Distance scores in the first round of testing for all twelve scenes. Shown in this graph are scores of the solutions of the three successful evolutionary runs using six scenes.

Three out of the ten evolutionary runs using only six-scenes, provided solutions which could solve the three basic mono-colour scenes (road brighter than non-road) and all six dual-colour scenes (Figure 3-5). This included scenes 10, 11 and 12 which they had not experienced during evolution. This is proof of the flexibility and adaptability of the solutions evolved. Not surprisingly they failed in the three reversed mono-colour scenes as the entire basis of their learning was dependent on the road being brighter than the non-road. On investigating the dynamic colour perception strategies of these controllers it was observed that in the three mono-colour scenes, the colour parameter ($\rho$, $\gamma$, $\beta$) corresponding to the non-random channel was steady and activated above 0.85 for majority of the trial duration. This was expected given their $\Delta$ values (see Section 3.1.5) from evolution being in the range of 1.4-1.8. However in places where sharp turns or course corrections were needed, a different behaviour was observed. The colour nodes instead of remaining at a constant high-activation value, oscillated between 0-0.9 every two time steps. It is also interesting to note that the motion in terms of dynamics was 'smoother' and faster when the 'correct' colour output was constantly activated to a high value ($\approx 0.9$). During the oscillatory

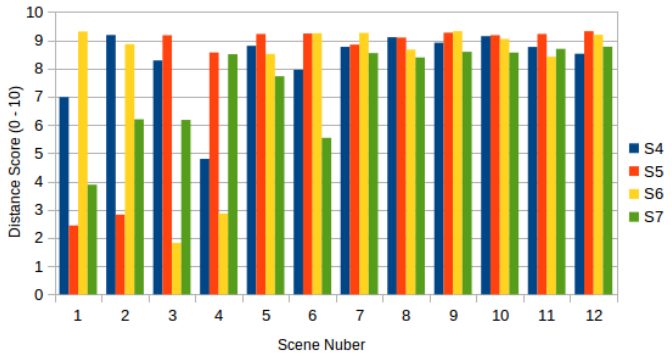



Figure 3-6: Distance scores in the first round of testing for all twelve scenes. Shown in this graph are scores of the solutions of the four best evolutionary runs using twelve scenes.

phases the motion was slower and more uneven, with regular course-corrections having to be made. Refer to Section 3.6 for a more comprehensive analysis of the neuro-controllers' active colour perception mechanisms, including the above mentioned sequences of oscillatory activation.

Results from the twelve-scene experiments (Figure 3-6) were not as uniform, with solutions showing greater variability in their colour perception strategies, depending on the seed and colour distribution set they were evolved in. The majority of solutions (like S5 and S6) only evolved the ability to dynamically vary two of their three colour outputs and simply did not use the third. This meant that two out of the six mono-colour scenes (basic and reversed) could not be solved. The unused colour output varied with each solution. However they did manage to solve all six dual-colour scenes because having the ability to dynamically vary only two colour outputs would be sufficient for these cases.

Only two solutions successfully evolved to show capability of solving all twelve scenes. Of these S4 evolved in scenes with colour distribution of Set *B* and S7 with distribution values of Set *A*. It is interesting to note the effect of these distribution values on the evolved solutions. The seed for S4 when used to evolve a solution with contrast values of Set *A* could develop only a sub-par solution where the controllers could not navigate the green mono-colour scenes. The seed for S7 when used with Set *B*, which could be said to be a less challenging environment, could only solve two scenes. Also unsurprisingly none of these seeds when tried with Set *C* could produce any solutions, as the contrast values were much

lower and the distributions themselves were more spread out across the intensity spectrum.

Solution S7, developed a strategy wherein their ability to differentiate on the basis of the green channel was more enhanced than the other two channels. The $\gamma$ output was constant and near maximum for all scenes where bright green could be made the differentiating channel. For all other scenes, the colour outputs oscillated between high and low activations every third time step. While the controllers did traverse the entire course in scenes 1 and 2, the navigation was slower and often error-prone at the beginning, contributing to the lower average scores. Solution S4 evolved behaviour where the $\rho$, $\gamma$ and $\beta$ terms were near maximum for the majority of the time for scenes 3, 8 and 9 respectively. In the rest of the scenes periods of both stable and oscillatory activations of the colour output nodes were observed.

**Test 2**

Four individuals, two each from the two best six-scene and twelve-scene runs, were then chosen to be subject to a further round of post-evolution testing in virtual roads. The aim of this round was to investigate the robustness of their road-following strategies by observing their behaviour in environments they had not encountered during the evolutionary phase. The twelve scenes were recreated with textures having average contrast of 90 and deviations from mean of around 40 (on a scale of 0-255). In each of these scenes, the range of distribution of the random noise channels was set at 0-0.80 for one case and 0-0.25 in another. In the evolutionary runs, the distribution of the random noise channels always varied from 0-1 with uniform probability. However it was observed that narrowing this range to 0-0.5 during the testing phase caused a few randomly selected controllers to fail and thus it was decided to add this as a further evaluation parameter. In theory, controllers with the correct feature extraction strategy would be able to completely discard the random channels, as despite the range of the distribution it had no contribution towards highlighting the desired features. The road was set to be of the “S" shaped type with an angle of curvature of 25° in both left and right initial starting directions. These shapes were generated twice, giving a total of 4 trials for each random noise distribution value in each of the 12 scenes. Thus each individual in this second round of testing was evaluated for 96 trials. In order

to further enhance the effect of presenting an unfamiliar environment to the controllers the road tile used in this testing phase represented a more delineated and unstructured course, having a maximum width of 110 cm at places but with only 85-90 cm consistently visible throughout. Figure 3-7 shows the distance scores of the two best solutions of this round, averaged across eight trials for each scene.

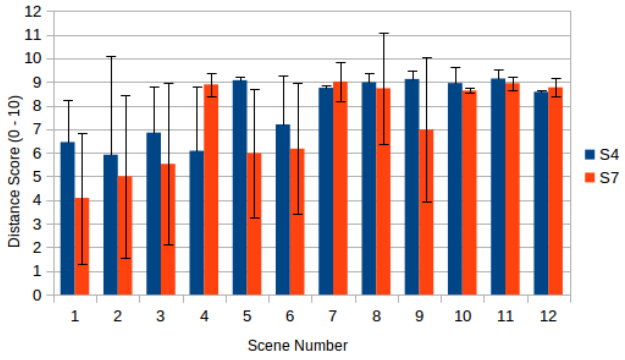



Figure 3-7: Average scores received by the two best solutions in the second round of testing. The error bars depict the associated standard deviation values.

The results of this second round of testing (Figure 3-7) showed that solutions S4 (twelve scenes) and S7 (twelve scenes) had developed the most robust and general-purpose solution. Despite receiving lower scores (below 7) for a few scenes, only these solutions had the capability of solving all twelve scenes across all the evaluation parameters, i.e all road shapes with reduced contrast and varying random noise values. The performance of S1 (six scenes) in identifying features in the blue or $\gamma$ channel was affected by the reduced contrast in the colour distribution. This in turn not only meant failure in the corresponding mono-colour scenes but also in the two dual-colour scenes where the blue channel was brighter on the road. The other two channels could still be successfully used across both ranges of the random noise variation. It was later tested in a scene with average contrast for the blue channel at 109 (still a new environment), and in this case it was able to navigate the corresponding scenes successfully.

While the solution S2 was able to solve almost all scenes when the random noise was in the range of 0-0.80, it failed to differentiate on the basis of both blue and green channels when this range was reduced to 0-0.25. This resulted in lower average scores for scenes 1, 2,

6 and 12. The inability to perform in these scenes was because it incorrectly associated the low distribution range of random values in the red channel with the availability of features. Thus it could see no contrast between the road and non-road surfaces in those scenes where red was not a feature differentiating channel.

For the two successful solutions in this round, it can be seen that in both cases performance in all but one mono-colour scene deteriorated compared to the earlier round of testing. They were still capable of reaching the end of the road in these scenes, but with less consistency compared to the earlier tests contributing to the lower overall score. Interestingly despite being subject to higher contrasts than S7 during evolution, S4 was still able to match or exceed its performance in eleven out of twelve scenes. On observing the behaviour of the controllers in these lower contrast scenes, it was seen that there was a disparity in their sensitivity to the three colour channels. For each solution there was a particular colour channel in which the ability to perceive contrast was much more pronounced. The controllers changed their colour perception strategies in these scenes, relying increasingly on oscillating the activations of the colour output neurons. However when the channel they were most sensitive to was available, they used it exclusively by activating only the associated output neuron for the majority of the trial.

## 3.3 Indoor Pioneer Trials with RGB

This section contains results from an initial round of robot-control trials carried out in road-segments created in an indoor laboratory. Observations from these experiments proved the viability of controllers generated through the methodology described in Section 3.1 for guiding a real mobile platform, as well as leading to the exploration of alternate illuminant-invariant colour combinations (beyond *RGB*) in subsequent experiments. It should be noted that the evolutionary run that generated the network chosen for these trials is different from that used for the virtual and outdoor trials in Sections 3.4 and 3.5. Due to hardware issues, operation in outdoor roads requires an update time of 0.38 seconds (as opposed to 0.10 seconds in the indoor laboratory). Thus a different set of evolutionary runs was carried out with the update time of 0.38 seconds to generate controllers for outdoor operation.

We ran a set of 15 differently seeded evolutionary simulations (runs), each of which lasted 4,500 generations. At the end of the evolutionary phase, we performed a first post-evaluation test in simulation. The aim of this post-evaluation test is to generate a more accurate estimate of the effectiveness of the most promising solutions in a larger set of operating conditions in order to select the best candidate to be ported on the real robot. For each evolutionary run, the best evolved controller or 'solution' of each of the last 2,500 generations is subject to a further round of re-evaluation. Each solution is re-evaluated on a set of 312 trials which included the original scenes illustrated in table 3.1, as well as three additional scenes rendered using different textures representative of real world environments (e.g., grass, asphalt). Each scene is presented multiple times by varying the shape of the road as well as the inherent contrast and distributions of intensities in the three RGB color channels. The 'genotype' corresponding to the neural network that drove the most successful robot (i.e., the one with highest number of roads navigated from start to end without going off-course) was chosen as the 'best solution' or controller to be ported and tested on the Pioneer robot. The tests with the real robot have been carried out in an indoor environment equipped with a motion capture system to track the robot movements. At this stage of the project, the indoor environment turned out to be more appropriate to our goal because it allows us to easily vary the robot operating conditions, such as the shape and the color composition of the road and non-road surfaces, and track the exact position of the robot to millimeter-level accuracy. Moreover considering the simplistic nature of the virtual evolutionary scenes, the indoor roads generated with combinations of different materials as well as general environmental features such as lighting and the presence of 3-D objects (e.g. chairs, tables, walls etc.) provided challenging test conditions. The different materials used to generate road/non-road combinations also correspond to unique noise profiles for the robot kinematics, due to differences in terrain roughness/friction. During testing, the best evolved neural network is ported on the real Pioneer robot, and a total of 120 trials has been executed in four different types of road/non-road conditions: **NF1**, **NT1**, **CF1**, and **CT1**. Fig. 3-9 shows the RGB color histograms for each condition. Eight further operating conditions have been "artificially" created by changing the pairing between $\rho$, $\gamma$ and $\beta$ and the mean value of each RGB color channel in the equation used to compute the

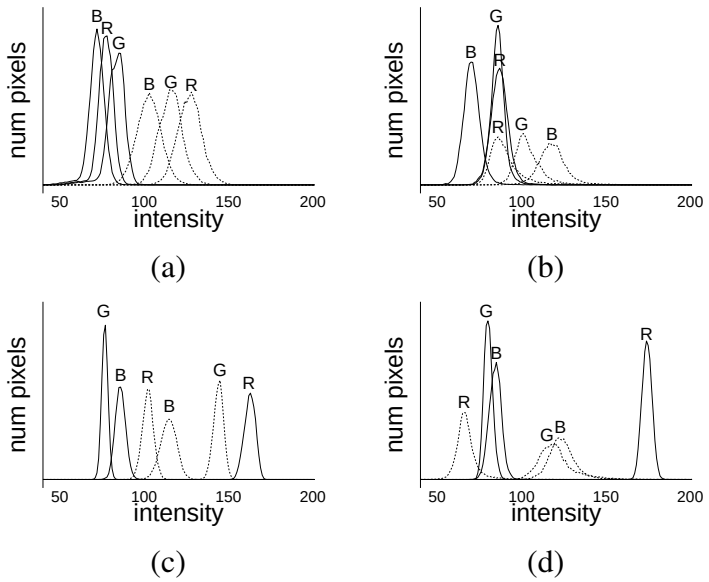
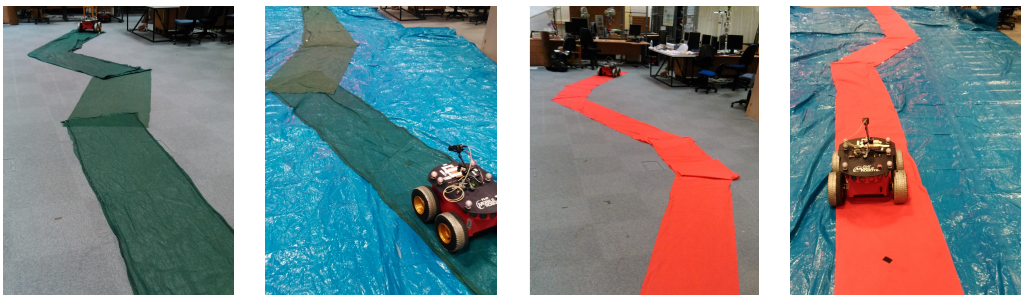



Figure 3-8: RGB color histograms for each condition: (a) **NF** condition: the road is a green mesh, and the non-road is the lab gray floor; (b) **NT** conditions: the road is a green mesh and the non-road is blue tarpaulin; (c) **CF** conditions: the road is a red carpet and the non-road is the lab gray floor; (d) **CT** conditions: the road is a red carpet and the non-road is blue tarpaulin. In each graph, continuous lines refer to the road and dotted lines refer to the no-road surface. The letters R (Red), G (Green), and B (Blue) indicate the color channel.

network input vector. The 'artificially' created operating conditions are generated with the following procedure. For each of the four basic conditions, we have first identified the color channel with the maximum amount of inherent contrast. Then, for each condition, the two extra variants have been generated by pairing the color channel with the maximum amount of inherent contrast with the other two color parameters not originally paired with it. For example, in condition **NF**, Red is the channel with the maximum amount of contrast. Thus, Red, paired with $\rho$ in **NF1**, is paired with $\gamma$ in **NF2** (i.e., $I_i = \rho \times C_B + \gamma \times C_R + \beta \times C_G$), and



(a) (b) (c) (d)

Figure 3-9: (a) nf, (b) nt, (c) cf, (d) ct

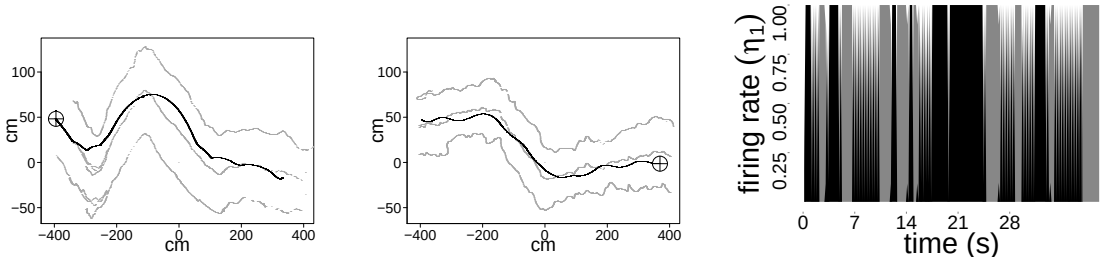
with $\beta$ in **NF3** (i.e., $I_i = \rho \times C_G + \gamma \times C_B + \beta \times C_R$) (see table 3.5, columns 7, 8, and 9 for the color parameters to colors pairing in the other conditions). Since for the evolved controllers $\rho$ is paired with Red, $\gamma$ with Green, and $\beta$ with Blue, the re-arrangement of the color parameters to colors is equivalent to testing the robot in a different environment, in which Red is whatever non-Red channel paired with $\rho$, Green is whatever non-Green channel paired with $\gamma$, and Blue is whatever non-Blue channel paired with $\beta$. We did not investigate all possible permutations per condition because this would have been too time consuming and possibly not particularly informative. For each of the 12 conditions, the robot undergoes 10 trials, 5 on a road featuring a right turn followed by a left turn, and 5 on a road featuring a left turn followed by a right turn. The road shapes is generated to ensure that in each of them the robot has to make at least one sharp turn in either direction. Table 3.5 shows the number

Table 3.3: Summary of the robot performance. Columns 2 and 3: mean and standard deviation of the road width. Column 4: number of successful trials per condition. Columns 5 and 6: mean and standard deviation of the robot divergence from the road center. Columns 7–9: color parameters to colors pairing, with the letter in bold indicating the RGB channel with the maximum amount of contrast.

| Cond. | Road (cm) | | Succ. | Deviation (cm) | | Col. Param. | | |
|---|---|---|---|---|---|---|---|---|
| | mean | sd | num. | mean | sd | $\rho$ | $\gamma$ | $\beta$ |
| NF1 | 84.51 | 10.27 | 10/10 | 10.01 | 1.70 | **R** | G | B |
| NF2 | 84.51 | 10.27 | 10/10 | 11.87 | 3.22 | G | B | **R** |
| NF3 | 84.51 | 10.27 | 10/10 | 11.99 | 5.17 | B | **R** | G |
| NT1 | 80.09 | 9.94 | 10/10 | 14.56 | 4.78 | R | G | **B** |
| NT2 | 80.09 | 9.94 | 8/10 | 7.70 | 4.02 | R | **B** | G |
| NT3 | 80.09 | 9.94 | 8/10 | 9.35 | 3.52 | **B** | G | R |
| CF1 | 73.11 | 9.32 | 9/10 | 15.42 | 12.85 | R | **G** | B |
| CF2 | 73.11 | 9.32 | 10/10 | 8.51 | 1.70 | R | B | **G** |
| CF3 | 73.11 | 9.32 | 10/10 | 13.44 | 8.06 | **G** | R | B |
| CT1 | 64.50 | 9.01 | 8/10 | 9.95 | 5.70 | **R** | G | B |
| CT2 | 64.50 | 9.01 | 6/10 | 10.60 | 9.75 | G | B | **R** |
| CT3 | 64.50 | 9.01 | 9/10 | 10.76 | 4.10 | G | **R** | B |

of successful trials during evaluation in each condition (Table 3.5, column 4). Note that a trial is considered successful if the robot navigates the road from start to end without left or right wheels going completely beyond the road boundaries. The robot succeeded in 108 out of the 120 evaluation trials with an overall success rate of 90%. Across all the environments, except **CT2**, the maximum number of failures was 2 which indicates that the robot is able

to comprehensively solve these scenes. In the 12 unsuccessful trials, we calculated that the robot managed to successfully traverse an average of 74.5% of the road length before going beyond road boundaries. On examining the trajectories of the failed trials in **CT2** where the robot displayed it's worst performance, 2 out of the 4 trials could be classed as complete failures. In the other two, the robot did display road-following behavior by executing both turns. However this was along the road edges, with the right wheels straying completely on to the non-road area. Thus, while trials could not be classified as being successful, roads in environment **CT** did have the least width (see Table 3.5, column 2 and 3) making such deviations more likely. It is worthwhile noting that for all the successful trials the maximum average deviation from the middle of the road is less than 16cm (Table 3.5, column 5). This is indicative of the robot's ability to maintain its course in the middle of the road and not deviate towards the edges. The trajectories in two successful trials can be observed in Fig. 3-10(a) and 3-10(b). Given the fact that the road length and width as well as the number and




(a) (b) (c)

Figure 3-10: (a) and (b) robot trajectory in conditions **NF1** and **CT1**, , which the boundaries and middle points of the road. Black circles indicate the starting position. (c) Activation over time of the color parameters $\rho$ (light gray), $\gamma$ (black) and $\beta$ (dark gray), during one trial in condition **NF1**.

type of turns were approximately kept similar between conditions, any difference between the trajectories of the robot in different conditions can be accounted for by referring to the robot capability to dynamically adjust its color perception system through variation of $\rho$, $\gamma$, and $\beta$. A close look at the way in which all the best evolved controllers vary the color parameters unveils complex patterns of variation which are hard to interpret. Fig. 3-10(c) shows the activation over time of the color parameters $\rho$ (light gray), $\gamma$ (black) and *beta* (dark Gray), during one trial of the real robot in condition **NF1**. In the other trials in

same and in different conditions we observed similar patterns to those shown in Fig. 3-10(c). The graph shows that the robot does not vary the parameter in the most intuitive or expected way: that is, by fixing to 1 the parameter corresponding to the color that returns the highest contrast between the road and non-road surface, and to 0 the other two parameters. Instead, what is observed is an irregular oscillatory behavior between the low and the upper bounds of the range, of primarily two of the three parameters, with the third one tending to oscillate only sporadically while remaining most of the time close to zero. These specific patterns of oscillation are different for each trial and they definitely bear a great significance for the development of successful navigation strategies. We found that, in all operating conditions, the robot continuously adjusts its perceptual system by exploiting two of the three color components of the visual scene. These components are temporally mixed in a complex way and further investigation in Section 3.6 provides a better understanding of the significance of these oscillatory patterns. Nevertheless, an initial qualitative analysis of the system provides enough evidence to claim that the system adapts its perceptual strategy to the color characteristics of the environment. The system adaptiveness can be more clearly seen in Fig. 3-11, which shows the distribution of values of the color parameters for all 10 trials in each condition with the real robot. The graph shows that different patterns of activation are generated for each of the four types of environments. In particular, we see that the medians (i.e., black horizontal bars within the boxes) of the distributions of the three parameters differ in most of the conditions. For example, in conditions **NF1**, **NT1**, and **CF1** $\rho$ varies more than in condition **CT1** where the median is zero. Even within each type of condition, the re-arrangement of the pairing between color channels and color parameters induces the robot to adjust its perceptual strategy. For example, in conditions **CT1** and **CT2** $\gamma$ varies more than in condition **CT3** where the median is zero. In condition **NT2**, $\beta$ varies more than in conditions **NT1** and **NT3** where the median is zero. Not much variability is observed in condition **NF** where the difference in intensity between the same colors in road and non-road surface is similar (see fig. 3-9(a)). Thus, it is likely that, in this condition, any change in perceptual strategy would not provide any further benefit in term of discriminating the road from the non-road surface. The graphs in Fig. 3-11 also shows that the robot tends to rely mostly on two of the three color parameters (i.e., $\rho$ and

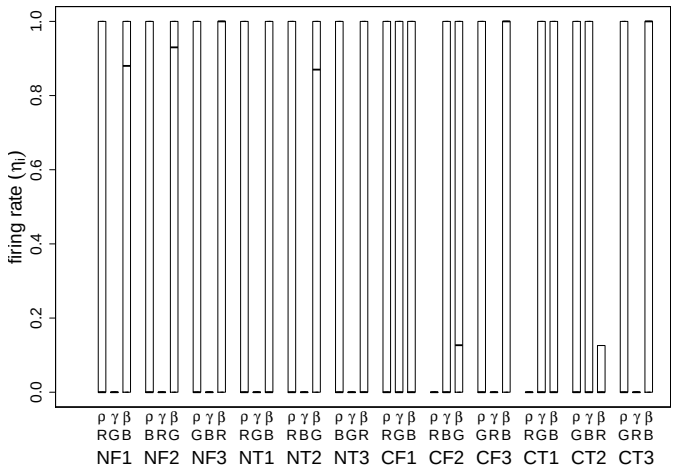


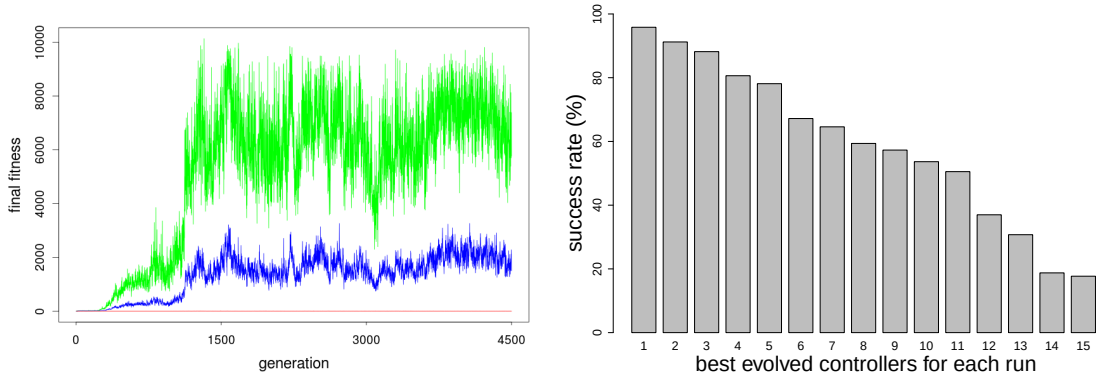

Figure 3-11: Box-plots showing the values, recorded during 10 trials in each condition, of the color parameter indicated on the x-axis. Below each color parameter is indicated the respective paired color (with R for Red, G for Green, and B for Blue). Boxes represent the inter-quartile range of the data, while black horizontal bars inside the boxes mark the median values. Instances where bars occupy almost the full range but have medians close to 0, indicates that the colour parameter activation oscillates between high and low values, albeit remaining low for the majority.

$\beta$). This preference is not entirely a consequence of the characteristics of the environment in which the robot was tested. Other controllers (including those discussed previously in Section 3.2),have been observed to have similar properties, with a tendency to use two out of the three color parameters. If these two color parameters do not generate the desired visual cues the robot starts varying the third one.

## 3.4 Virtual Trials exploring colour models

To generate a controller that can be the basis of trials in more complex outdoor roads, we ran a set of 15 differently seeded evolutionary simulations (or evolutionary runs), each of which lasted 4,500 generations. At the end of the evolutionary phase, similar to the experiment described in Section 3.3 a first post-evaluation test in simulation is carried out. The aim of this post-evaluation test is to generate a more accurate estimate of the effectiveness of the most promising solutions in a larger set of operating conditions, and to choose what will be our best evolved controller. For each evolutionary run, we post-evaluated the best solution

(i.e., the best genotype based on fitness) of each of the last 2,500 generations. Each solution has been re-evaluated on a set of 192 trials based on the 12 evolutionary scenes described above. Each scene has been presented 16 times by varying the shape of the road. We use 8 different road shapes, as well as two different intensity distributions of the inherent contrast in colour channels. The results of this first post-evaluation tests are shown in Figure 3-12, which shows the percentage of success of the best evolved controllers of each evolutionary run, during the 192 post-evaluation trials[1]. The graph shows that 4 out of 15 evolutionary runs managed to produce controllers with a success rate higher than 80%. The controller that drove the most successful robot (i.e., the one with highest number of roads navigated from start to end without leaving the road) has been chosen to be our best evolved controller.




(a) (b)

Figure 3-12: (a) Fitness graph for best evolutionary run. Green indicates the best, blue the average and red the lowest fitness in a generation. (b) Percentage of success of the best evolved controllers of each evolutionary run, during 192 trials (12 scenes, 8 road shapes, 2 distributions of colour intensity). Controllers are ranked from best to worst.

In the rest of this section, we show the results of a series of analyses in which the best evolved controller (hereafter, B-controller) has been tested on a larger set of simulated operating conditions generated by varying the characteristics of the scenes as well as the nature of the colour components used to generate the controller sensory input. For this purpose we carried out five different tests, that feature various forms of colour variability between and within scenes. These series of tests aim to verify the robustness and adaptability of the B-controller. In the following Section 3.5, the capability of the B-controller to cross the reality gap is demonstrated by showing the results of a series of tests in which the controller

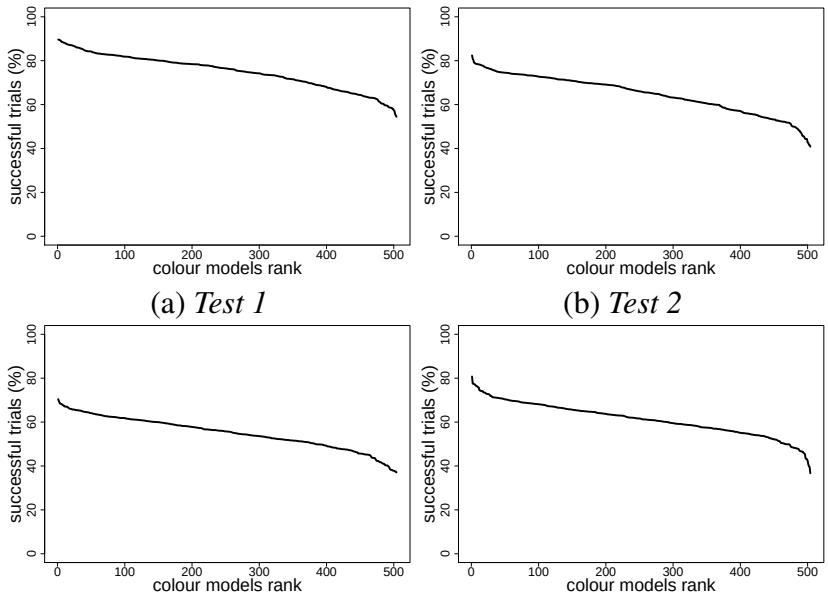
[1]Links to videos of the robot controlled by the best evolved controller are available in Appendix A.

guides a real Pioneer robot in 5 different outdoor environments.

In *Test 1* to *Test 4*, the B-controller is evaluated under conditions in which the sensory inputs $I_i$ (Equation 3.1) are computed by considering all the possible 3-permutations of the following 9 colour components: R, G, B, H, S, Y, U, a, and b (respectively from the RGB, HSV, YUV and CIE $L^*a^*b^*$). Refer to Appendix B for further details on the conversions of these colour models from the standard RGB representation. Hereafter, each arbitrary triplet of colour components is referred to as a colour model. Triplets made of same colour components, but ordered differently, differ in the way in which the colour components are associated to the colour parameters $\rho$, $\gamma$ and $\beta$. The logic behind the use of multiples colour models is the following: a successful controller that adaptively varies the colour parameter $\rho$, $\gamma$ and $\beta$ may benefit from the use of any arbitrary combination of three components chosen from different colour spaces. For example, with the use of components of colour spaces different from RGB, in which colour and luminance are not intertwined, a successful adaptive controller may more effectively respond to extra environmental variability, or dynamic variations produced by shadows, sudden changes in luminance, etc.[2]

*Test 1* aims to evaluate the B-controller on a large set of scenes (812) generated by the 2-permutations of 29 different textures. In each scene, one texture is used for the road and the other for the non-road surface. Given that each scene is presented with 8 different road shapes, and associated to the 504 colour models given by permuting the 9 colour components R, G, B, Y, U, a, and b, the total number of trials is $812 \times 8 \times 504$. The purpose of *Test 2*, *Test 3*, and *Test 4* is to evaluate the performance of the B-controller in scenes in which there is more variability in term of colour distribution compared to scenes presented during the training phase, since each scene features three (instead of two) different textures. In *Test 2*, the texture of the two non-road areas, on the left and on the right of the road, are different. In *Test 3*, left and right non-road areas are the same, but the non-road texture on both sides of the road changes colour half-way through the road. In *Test 4*, left and right

[2]During evolution, the RGB colour space has been used to create the scenes and also to generate the input vector of the robot controllers. The reason for this is, first, that it is relatively simple altering R, G, B of an image texture rather than components of different colour spaces to generate the environmental variability required to select controllers capable of adaptively tuning their perception system (i.e., the $\rho$, $\gamma$ and $\beta$) to different scenes. Second, it would have required longer in term of computational time, to evaluate controllers on the use of different components of multiple colour spaces.




(a) *Test 1* (b) *Test 2*

(c) *Test 3* (d) *Test 4*

Figure 3-13: Results of *Test 1*, *Test 2*, *Test 3*, *Test 4*. Each graph shows the percentage of successful trials for all 504 colour models. In each graph, the colour models are ranked from the best to the worst.

non-road areas are the same, and the road texture changes colour half-way through the road. We created 720 scenes, resulting from all possible 3-permutations of 10 different textures. Each scene is presented with 8 different road shapes, and it is repeated 504 times, one for each colour model given by permuting the 9 colour components R, G, B, Y, U, a, and b. It follows that the total number of trials in *Test 2*, *Test 3*, and *Test 4* is $720 \times 8 \times 504$.

As during evolution, in all tests, the robot controller is reset at the beginning of each trial. A trial is considered successful if the robot manages to traverses all 11 tiles forming the road. A trial is terminated earlier and set as unsuccessful if the robot is detected to have moved off the road. The results of *Test 1* to *Test 4* are shown in Figure 3-13, where each graph indicates the percentage of successful trials of all 504 colour models. We also compare the colour models by considering these four tests as a multi-objective optimization problem, in which each model has to maximize its performance measured in terms of percentage of successful trials. Table 3.4 shows, for each test, the performance of the robot in trials in which the controller is linked to those colour models populating the Pareto set, plus the RGB colour model. For each test, the colour models are ranked in descending order of performance. There are various lessons to learn from this data.

Table 3.4: Performance of the colour models in the Pareto set, and RGB, for *Test 1*, *Test 2*, *Test 3*, *Test 4*and *Test 5*. For each test, the colour models are ranked in descending order of performance.

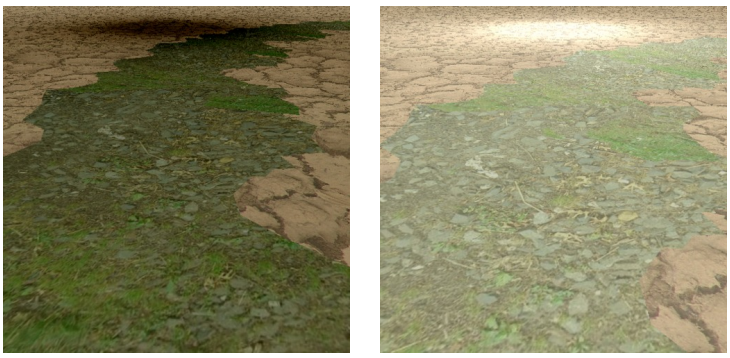
| *Test 1* | | *Test 2* | | *Test 3* | | *Test 4* | | *Test 5* | |
|---|---|---|---|---|---|---|---|---|---|
| **colour model** | **Success rate (%)** | **colour model** | **Success rate (%)** | **colour model** | **Success rate (%)** | **colour model** | **Success rate (%)** | **colour model** | **Success rate (%)** |
| VHb | 89.63 | bSH | 82.41 | bUV | 70.45 | bUa | 80.76 | bUV | 75.25 |
| VHa | 89.51 | aSH | 80.97 | UbV | 69.56 | Uba | 77.51 | bHV | 75.25 |
| bHU | 89.08 | HbB | 79.06 | aSH | 68.83 | Bha | 77.50 | bHa | 73.25 |
| bHA | 88.59 | USH | 78.07 | VHb | 68.35 | VHa | 77.36 | VHa | 73.25 |
| aSH | 88.51 | bHV | 76.87 | UbB | 68.26 | UbV | 77.18 | VHb | 71.25 |
| aHB | 88.40 | UbB | 76.33 | HbB | 66.54 | bUV | 76.82 | aHb | 69.75 |
| bHV | 88.06 | bHa | 76.07 | aHB | 65.72 | aHb | 76.49 | bUa | 68.25 |
| bUA | 87.83 | bHU | 74.53 | USH | 65.71 | VHb | 75.88 | UbV | 67.00 |
| UbV | 87.74 | VHb | 74.44 | bHV | 65.67 | UbB | 74.16 | Uba | 67.00 |
| bUV | 87.74 | bUV | 73.26 | Uba | 63.26 | bHV | 72.74 | bHU | 63.75 |
| Uba | 87.11 | bUa | 73.26 | bHU | 62.46 | HbB | 71.47 | USH | 63.50 |
| USH | 86.79 | aHb | 72.72 | bUa | 61.85 | bHU | 71.23 | aSH | 62.00 |
| UbB | 85.79 | VHa | 72.43 | bSH | 59.37 | aSH | 70.12 | bSH | 60.25 |
| HbB | 81.86 | UbV | 70.67 | bHa | 58.24 | USH | 69.84 | UbB | 33.75 |
| bSH | 78.34 | Uba | 68.83 | VHa | 53.55 | bSH | 65.93 | HbB | 31.50 |
| RGB | 57.00 | RGB | 52.22 | RGB | 45.70 | RGB | 54.00 | RGB | 13.25 |

We consider first *Test 1* and *Test 2* where the colour variability is only between the scenes and not within each scene. Both tests evaluate the robustness of the B-controller to deal with a larger number of road/non-road textures combinations never encountered during training. From Table 3.4, it can be seen that in *Test 1* the success rate of the robot with almost all Pareto set colour models is higher than 85%, while in *Test 2* the robot success rate with at least half of the Pareto set colour models is higher than 75%. These results indicate that, if supported by adequate colour models (i.e., models which can potentially generate the perceptual cues required to visually discriminate the road from the non-road surface in such variable operating conditions), the B-controller manages to successfully cope with a large range of different scenes, proving to possess the required robustness to deal with scenes never encountered during training. Although the data gathered from these tests do not tell us anything about how $\rho$, $\gamma$ and $\beta$ are varied between the scenes, the relatively high success rates under these testing conditions suggest that the controller copes with the colour differences of the scenes by adaptively varying the colour parameters $\rho$, $\gamma$ and $\beta$ to

exploit the benefits of those components of each colour model that facilitate the navigation task. The results of *Test 1* and *Test 2* also show that not all colour models adequately support the controller in the navigation task. The graphs in Figures 3-13(a) and 3-13(b) show that for many colour models the robot success rate drops below 70%. This suggests that the adaptiveness required to cope with the environmental variability encountered by the robot in these tests is a combination of the functional properties of the controller and the characteristics of the colour model. Each colour model offers a unique perspective on the scenes from which the controller has to extract the required perceptual features to guide the robot in this navigation task. Quite surprisingly, RGB, which is the model used during the evolutionary phase, is not among the models that allow the robot to perform reasonable well at both tests. As expected, although perfectly adequate, the performance observed in *Test 2* is worse than that observed in *Test 1*. Indeed, *Test 2* uses scenes where right and left non-road surfaces have different colour, a situation not encountered during training.

While *Test 1* and *Test 2* feature variability between the scenes, *Test 3*, and *Test 4* are definitely more challenging since they feature colour variability between and within the scenes. Recall that the controller is reset only at the beginning of each trial. Thus, in order to cope with variability within a scene, the controller has to adjust the colour parameters without being reset at the time when, during the trial, the environmental conditions change. The results of these tests show some positive aspects, and areas in which improvement is needed. The positive data are those of *Test 4*, where the robot's success rate with several Pareto set colour models is higher than 75% (see Figure 3-13(d) and Table 3.4). The B-controller can successfully cope with conditions in which the road texture abruptly changes, in spite of the fact that this event has not been contemplated during training. In *Test 3* the success rate is not as high as for the other previous tests. The change in colour texture of the non-road area seems to hinder the performance of the robot more than any other type of environmental variability (Figure 3-13(c)).

We also run a fifth test (*Test 5*) to evaluate the B-controller in scenes with shadows and bright spots. To keep the computational time required to run this test within reasonable limits, we used a limited set of 25 different scenes. In each scene, the colour of the non-road and of the road surfaces does not change but shadows and bright spots are added to



(a) (b)

Figure 3-14: Images of simulated environments used in *Test 5*, in which the scenes presents (a) shadows, (b) bright spots.

the scenes (see supplementary document for images of these scenes). As in previous tests, each scene is presented with 8 different road shapes. However, each scene is repeated for only the 15 colour models populating the Pareto set as resulted from *Test 1* to *Test 4*, plus the RGB colour model. The total number of trials in *Test 5* is therefore $25 \times 8 \times 16$. The results are shown in Table 3.4. We can see that, even in these very challenging conditions, the robot's success rate with at least two Pareto set colour models (i.e., bUV, and bHV) is higher than 75%.

In summary, given that the large majority of the scenes employed in these tests have been created with textures not used during training, the relatively good performances of the robot guided by the B-controller associated to several colour models bear upon the robustness of the controller. Moreover, these results demonstrate that the design method is particularly effective in generating adaptive mechanisms for solving this visual navigation task. The next interesting point emerging from the data is that there is no single colour model that performs better than all other models in all tests. The final point to highlight is that the RGB model is not the best performing colour model in any of the tests (see Table 3.4).

## 3.5 Outdoor Pioneer Trials

This section illustrates the results of tests in five different outdoor environments with the real Pioneer robot guided by the B-controller. The robot is tested in 5 different environments


Figure 3-15: Outdoor environments. In each image, the black line refers to the robot's trajectory for a trial. The green circle shows the starting position, and the red circle denotes the end of the trajectory. Width (m) and length (m) of each path is indicated above each image. Images in the topmost, middle and bottom rows correspond to trials carried out with the USH, ASH and BUV colour models (respectively).

(see Figure 3-15 for details). For each environment, the robot undergoes a total of 30 trials, as 3 sets of 10 trials with 3 different colour models (i.e., USH, aSH, bUV). We used these three colour models because, in previous tests, they proved to be able to adequately support the robot guided by the B-controller in a variety of simulated environmental conditions.

For each set of 10 trials, at the sixth trial the robot's starting position changes from the beginning to the end of the outdoor path, and consequently its direction of motion is inverted. A trial is successfully terminated when the robot traverses the entire length of the road without moving off the road boundaries. A trial is unsuccessfully terminated when either one set of wheels goes off the road boundaries, or for exceeding the time limits set to 5 minutes. Since Path 1, Path 2 and Path 3 are sections of a longer public path, they all shared the same road surface but they differ in terms of non-road surfaces on either side (see Figures 3-15(a), 3-15(b) and 3-15(c)). Extra variability not only between environments but also between trials in the same environment has been created by the varying weather and lighting conditions encountered during evaluation. Moreover, as all environments are public areas, pedestrians walking on the paths while trials were in progress represented a further challenge the robot had to deal with. The trials were not stopped for pedestrians walking in front of, or around the robot. These real environments are obviously more complex than those recreated in simulation and used for training, not only for the presence of pedestrians on the road surface, but also in terms of the colour distribution, varying lighting conditions, and for the presence of visual features such bushes and benches. Moreover, the width of the road of these outdoor environments is slightly wider than the surfaces of road in the simulated environments. The ability of the B-controller to deal with all these new elements should be considered further testament to its robustness.

The results of the outdoor tests are shown in Table 3.5. Considering all 150 trials across the three colour models, the overall success rate is 76%. We consider this as a good result that demonstrates the effectiveness of the evolutionary method in synthesizing a robust robot controller capable of successfully guiding a real robot engaged in this visual navigation task. Moreover, these results demonstrate the effectiveness of the embodied active approach, which generated a dynamic perception and action system capable of coping with the large environmental variability characteristic of the outdoor testing conditions.

Table 3.5: Number of successful trials per colour model for each outdoor environment (column 3); mean and standard deviation of the robot's divergence from the centre of the road (columns 4 and 5); mean and standard deviation of the time taken to complete a trial (columns 6 and 7). Negative divergence values indicate displacement leftwards of the road center.

| Env. | colour Model | Num. Succ. | Divergence (cm) | | Time (s) | |
|---|---|---|---|---|---|---|
| | | | mean | sd | mean | sd |
| Path 1 | USH | 8/10 | -9.4 | 39.1 | 168 | 66.1 |
| | aSH | 4/10 | -19.9 | 39.2 | 186 | 56.1 |
| | bUV | 8/10 | -43.3 | 74.6 | 190.5 | 15.8 |
| Path 2 | USH | 8/10 | -14.3 | 38.3 | 137.2 | 22.3 |
| | aSH | 8/10 | -18.9 | 42.7 | 168 | 45.0 |
| | bUV | 5/10 | 10.3 | 61.6 | 223.5 | 18.6 |
| Path 3 | USH | 8/10 | 4.4 | 41.1 | 127.8 | 33.6 |
| | aSH | 8/10 | -8.5 | 52.5 | 152.4 | 21.9 |
| | bUV | 10/10 | 7.2 | 33.9 | 231 | 25.2 |
| Path 4 | USH | 9/10 | -6.0 | 37.7 | 170 | 21.5 |
| | aSH | 4/10 | 3.4 | 34.2 | 136.5 | 22.5 |
| | bUV | 10/10 | -4.2 | 22.5 | 165.5 | 30.4 |
| Path 5 | USH | 9/10 | -10.5 | 108.3 | 102 | 17.5 |
| | aSH | 5/10 | -59.6 | 49.3 | 147.6 | 37.6 |
| | bUV | 10/10 | -14.0 | 68.5 | 226.8 | 54.3 |

Among the three colour models, bUV is the most successful one with an average success rate of 86%. However, bUV has a low success rate (50%) in Path 2. For all trials in which the B-controller has been associated to bUV, the robot systematically fails Path 2 for one of the two directions of motion, around the path half-way point which corresponds to a texture change on one of the non-road surfaces. When tested with the colour model USH, the robot achieves a slightly lower average success rate of 84%, but with a more constant performance across all five environments. The worst robot performance is with the colour model aSH with an average success rate of 58%. It should be noted that during the evolutionary design phase, no particular restrictions have been imposed either to the trajectory or to the speed of motion, apart from the requirements of navigating the paths without straying from the road boundaries, and of reaching the end within a given time limit (450 update cycles or 171 s). This latter requirement creates and implicit selective pressure favouring controllers that guide the robot at a speed sufficiently high to cover the 17.6 m of the simulated roads in less than 171 s.

During the outdoor trials, we captured the mean and standard deviation of the robot's divergence from the centre of the road. From the divergence values shown in Table 3.5, we notice that the robot controller associated with the colour model USH generates trajectories that are closer to the centre of the road than for the other two colour models. The use of USH helps the robot to develop a navigation strategy that tries to remain as far away as possible form the road edges. However, the relatively high standard deviation of the divergence for almost all colour models indicates that the trajectories have been rather sinuous, with quite a few changes in direction of motion to avoid to cross the road edges. For trials on Path 5 for all colour models, the robot tended to remain on the darkest lane (see Figure 3-15(e)). However, with the colour models USH and bUV, and only for one travel direction, the robot mainly uses the brightest lane (see in Figure 3-15e)), and it switches between lanes towards the end of the path. This accounts for the high standard deviation for the divergence from the road centre.

During the tests with real robots, no trial failed because of exceeding the time limit of 5 minutes. However, if we look at the average trial completion time (see Table 3.5, column 6) we notice that trials carried out with the bUV colour model took generally longer than trials with the two other colour models. We observed large variations in trial completion time between trials with any given colour model (see Table 3.5, column 7). This is caused by the behaviour of the robot which, in some trials, reduces its speed in certain sections of the path and displayed periods of rapid changes in direction without much progression along the path.

## 3.6 Analysis of the mechanisms underpinning adaptivity and robustness to environmental variability

Results from the previous sectons have demonstrated that the best evolved controller manages to successfully guide simulated and real robots in a variety of environments which differ in term of the color of the road/non-road surfaces. We showed that the controllers are robust enough to deal with environmental conditions never encountered during training.

From the robot performance and the kind of variability faced during post-evaluation, we deduced that the controller is able to tune the color parameters $\rho$, $\gamma$ and $\beta$ in order to adjust the visual input to the color characteristics of the environment in which the robot operates. In this section, we provide evidence of how the color parameters are actually varied by the best evolved controller.

We begin our investigations by showing the results of a qualitative test which visualizes the variation over time of $\rho$, $\gamma$ and $\beta$ while the simulated robot navigates four different environments (i.e., four trials) in which the color of the road surface changes half-way through the road. The simulated robot is controlled by the best evolved controller linked to the RGB color model. The results are shown in Figure 3-16, where each column refers to a different trial. Different combinations of colors are used for the road in each trial (Figure 3-16, first row).

The graphs in the top row of Figure 3-16 are a representation of the visual scenes experienced by the robot. The graphs are constructed by taking into account, at each update cycle of the robot controller, the contribution of $C_R$, $C_G$, and $C_B$ for each of the 25 grid cells superimposed on the camera image. At each update cycle, the 25 colored points are distributed over the y-axis. The graphs in the second row from the top refer to the variation over time of the 25 values sensory input vector (i.e., $I_i$, see Equation (3.1)). At each update cycle, the 25 gray scale points corresponding to the vector $I_i$ are distributed over the y-axis. The graphs in the third row from the top of Figure 3-16 refer to the variation over time of the color parameters $\rho$ (light gray), $\gamma$ (dark gray) and $\beta$ (black). The shades of gray indicate the activation of each color parameter at each update cycle. The graphs in the fourth row from the top of Figure 3-16 refer to the output to the left and right motors ($M^L$ and $M^R$ respectively).

Looking at the variation of the color parameters over time, it can be noticed that different types of activation patterns are used (e.g., see Figure 3-16, third row), and that the type of pattern tends to change when the color characteristics of the road surface changes (e.g., see Figure 3-16(b), at about 60 s). We notice that the controller employs three different types of activation patterns. We can observe an oscillatory pattern in which the controller alternates between high and low activation of all color parameters. This can be seen in Fig-

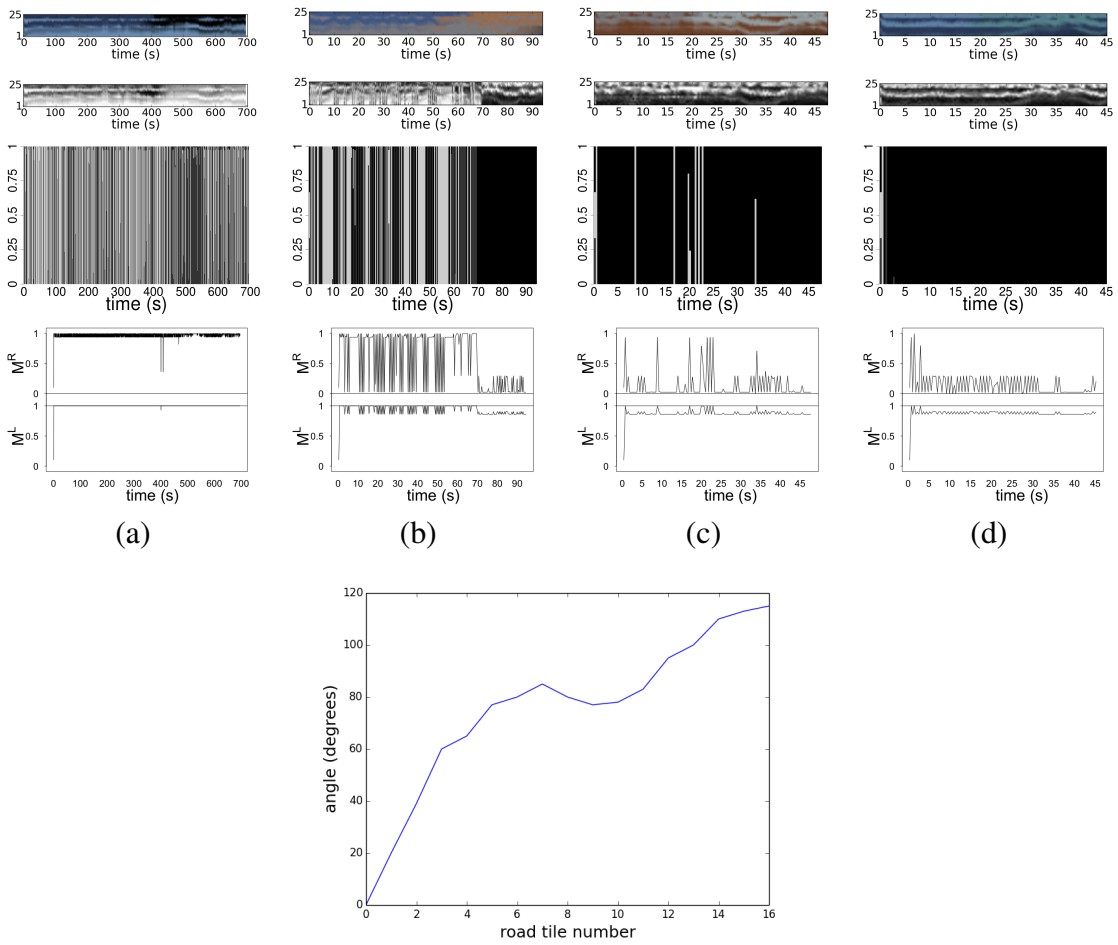



Figure 3-16: Color component activation. Each column of the top four rows refers to a different trial in a different environment, with the simulated robot controlled by the best evolved controller linked to the RGB color model. For each environment there was a change in road-surface texture roughly mid-way in the course, to gauge the effect of this sudden variation on the networks colour feedback node activations. All trials had a common road-shape with initial curvature towards the left. This shape can be inferred from the bottom most graph, which plots the cumulative curvature of the road, starting from the initial $20^o$ rotation to the left. The graphs in the top row are a representation of the visual scenes experienced by the robot. The graphs are constructed by taking into account, at each update cycle of the robot controller, the contribution of $C_R$, $C_G$, and $C_B$ for each of the 25 grid cells superimposed on the camera image. At each update cycle, the 25 colored points are distributed over the y-axis. The graphs in the second row from the top refer to the variation over time of the 25 values sensory input vector (i.e., $I_i$, see Equation (3.1)). At each update cycle, the 25 gray scale points corresponding to the vector $I_i$ are distributed along the y-axis. The graphs in the third row from the top refer to the variation over time of the color parameters $\rho$ (light gray), $\gamma$ (dark gray) and $\beta$ (black). The shades of gray indicate the activation of each color parameter at each update cycle. The graphs in the fourth row from the top refer to the output to the left ($M^L$) and right motors ($M^R$).

ure 3-16(a) third row, where in the first half of the trial there is a prevalence of $\rho$ (light gray), while in the second part of the trial there is a prevalence of $\gamma$ (dark gray), with spikes of $\beta$ (black) appearing throughout the trial. We can also observe an oscillatory pattern in which the controller mainly employs two color parameters, and oscillations tend to have a lower temporal frequency than the previous oscillatory pattern, for example in the first part of the trial in Figure 3-16(b) third row. Finally, we can observe a pattern in which one color parameter is set to its highest values while the other two are set to zero, Such as in the second part of the trial in Figure 3-16(b) third row, and in trials in Figures 3-16(c) and 3-16(d) third row, where $\beta$ dominates over $\rho$ and $\gamma$.

These three activation patterns of the color parameters are quite prototypical, since they exhaustively represent all the possible types of activation patterns observed in different environmental conditions, and with the controller linked to different color models. By looking at the variations of the color parameters in a larger set of simulation trials extracted from *Test 1*, *Test 2*, *Test 3*, *Test 4* and *Test 5*, described in Section 3.4, we noticed that changes in activation patterns of the color parameters are triggered by different types of environmental variability (e.g., changes in the color of the non-road surfaces, or appearance of bright spots and shadows), and that the oscillatory patterns are the most frequent ones. We believe that the oscillatory patterns of the color parameters are a rather effective exploratory strategy that allow the controller to tune the vision system to the characteristics of the environment, both during the initial phase of the trial, when the environment is unknown to the robot, and in response to environmental variations that require a change in activation pattern. Future work is required to test this hypothesis. On the basis of the activation patterns observed in Figure 3-16 we can say that a dynamic change in the environment tends to result into a change in the nature of temporal activation of the color parameters exhibited by the controller. If the controller cannot continue to extract a satisfactory final input vector after being exposed to a change in the environment, it changes the pattern of activation of its color parameters to maintain the robot navigation capabilities and to stay within the road boundaries.

To further demonstrate that the controller's color perception system is integral to for its ability to navigate and not simply a by-product of the artificial evolution, we consider the

Table 3.6: Table showing the relative performance in *Test 1*of the dynamic colour mixing neural network proposed by this paper. It is compared to a CTRNN using inputs when the dynamic colour-mixing is bypassed and high/low contrast colour channels are fed through.

| **Network** | **Colour Input** | **Percentage Successful Trials** |
|---|---|---|
| CTRNN 3 Layers | RGB (Dynamic Mixing) | 57.0 |
| CTRNN 3 Layers | RGB (Highest Contrast forced) | 41.5 |
| CTRNN 3 Layers | RGB (Lowest Contrast forced) | 35.5 |

network using the RGB color model but with the feed-back from its color outputs ignored. Instead based on analysis of the color-properties of the textures used to render scenes in *Test 1* we know the color channel carrying the highest/lowest contrast level for a particular environment. We feed the relevant highest/lowest contrast channel to the network for all the trials instead and cut-off the network's color perception feed-back. From the results it can be seen that the network performs the best (57.0 % success) when it's feedback properties are not interfered with. Receiving the highest possible contrast values as it's final input vector deteriorates the performance to 41.5 %.

Figure 3-16 also shows an interesting relationship between the perception and the motor system of the best evolved controller. The fourth row of Figure 3-16 shows that at the points where the road color changes there is also either a marked change (see Figure 3-16(b) and 3-16(c) fourth row) or a temporary disruption (see Figure 3-16(a) and 3-16(d) fourth row) of the nature of motor output activation. Trials with extended periods of color, and hence motor output, oscillatory patterns take significantly longer to complete. Trials in Figures 3-16(c) and 3-16(d) where there are only short periods of oscillatory behavior last 48 s and 45 s, respectively. This is in contrast to the trial in Figure 3-16(a) where the robot takes 698 s to cover the same distance (21 m). In the trial in Figure 3-16(b) the robot moves much slower in the first section of the road where the color parameters oscillate, compared to the second half of the trial where it covers the half-trial distance in approximately 20 s. This indicates that the controller's color perception and motor system are tightly coupled.

In the remaining part of this section, we show the results of further analysis that aims to

Table 3.7: Centroids of the three clusters resulting from the mean-shift sorting algorithm. Values in the cells are the percentage of time each color parameter is either below the low threshold of 0.2, or above the high threshold of 0.8.

| | Percentages (%) | | | | | |
|---|---|---|---|---|---|---|
| **Cluster** | $\rho > 0.8$ | $\rho < 0.2$ | $\gamma > 0.8$ | $\gamma < 0.2$ | $\beta > 0.8$ | $\beta < 0.2$ |
| one | 17.8 | 81.1 | 59.3 | 39.8 | 21.1 | 77.3 |
| two | 5.6 | 94.0 | 88.2 | 10.9 | 5.1 | 94.0 |
| three | 21.6 | 77.3 | 41.3 | 58.3 | 35.7 | 63.0 |

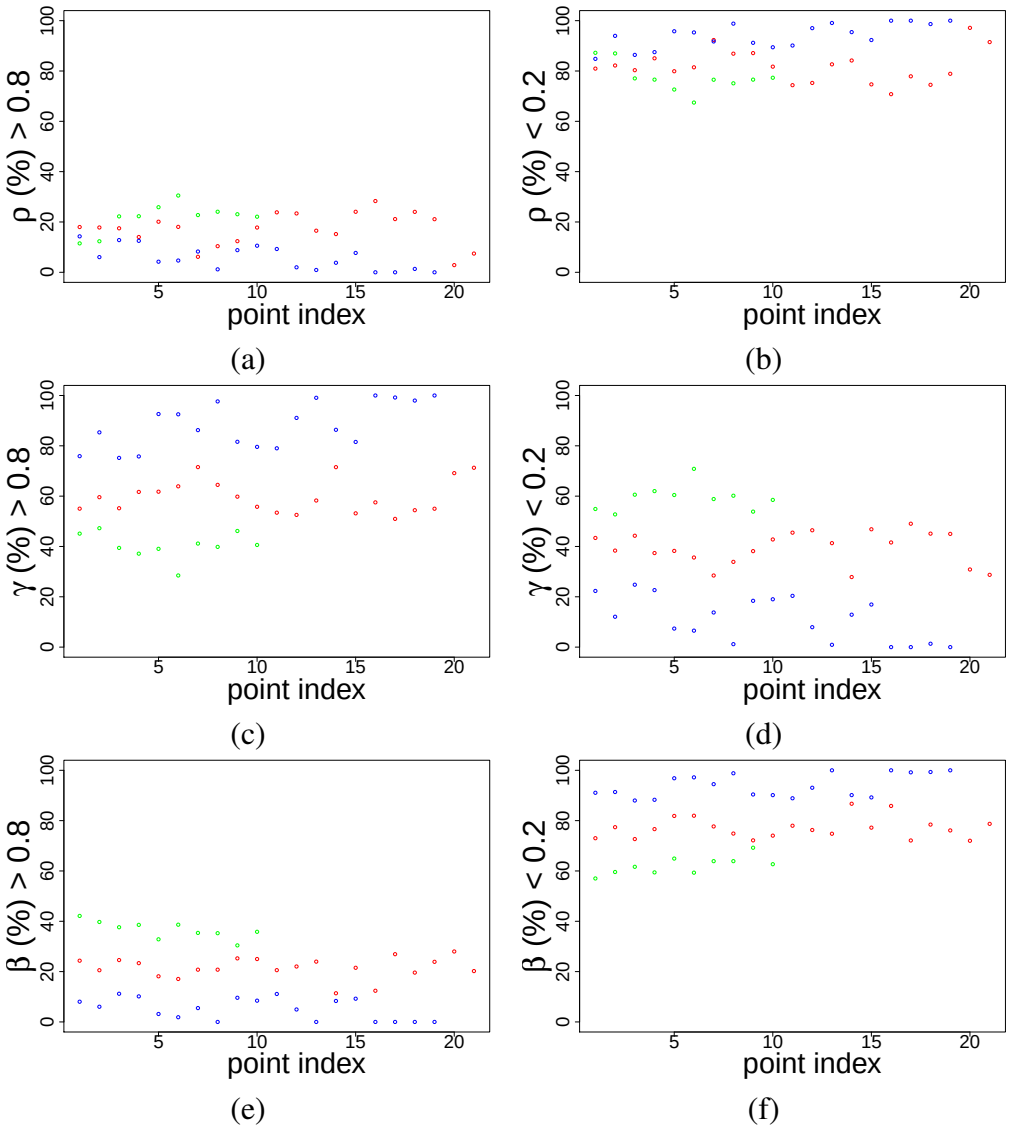



Figure 3-17: Components of the 6-dimensional points considered for clustering. Points in red, blue and green correspond to clusters one, two and three, respectively (see also Table 3.6).

quantitatively verify the hypothesis that the controller adapts to different environments by varying the activation of the color parameters $\rho$, $\gamma$ and $\beta$. For this we consider all 50 outdoor trials carried out with the USH color model, across all 5 environments shown in Figure 3-15. In each of these trials we recorded the values of the color parameters during the entire trial with the exclusion of the first 10 iterations, where color parameter are not set properly yet, and the last 10 iterations, where color parameters tend to be affected by the perception of the road end. Given that the previous qualitative tests with simulated trials indicated that the best evolved controller tends to develop oscillatory patterns in which each color parameter is either close to its maximum (1) or to its minimum (0), we classify the raw data as follows. We compute the percentage of time each color parameter is either below a low threshold of 0.2, or above a high threshold of 0.8. This reduces the raw data gathered from each trial to a point in a six dimensional vector space, where the six dimensions are given by the two types of categories (values either below the low threshold or above the high threshold), times the three color parameters $\rho$, $\gamma$ and $\beta$. We then clustered the 50 points obtained from all outdoor trials using the mean-shift unsupervised clustering algorithm (see [18]). The algorithm clustered the data into three clusters, the centroids of which are shown in Table 3.6. We observed that clusters one and three correspond to phases of trial where all three color parameters are activated in an oscillatory manner (see Table 3.6), while cluster two corresponds to phases of trial where the controller primarily keeps $\gamma$ set to 1. Figure 3-17 shows how the 50 points are distributed in the 6-dimensional space, and how they are clustered. The red points in Figure 3-17 correspond to cluster one, the green point to cluster two, and the blue points to cluster three.

Figure 3-18 shows the distribution of the points in the 6-dimensional space among the three clusters for each outdoor environment. The graphs clearly indicate that the best evolved controller tends to use different activation patterns in different environments. For example, while in Path 1and in Path 4 the activation patterns corresponding to clusters one and three are more represented than the activation pattern corresponding to cluster two. In Path 2, Path 3 and Path 5 the activation patterns corresponding to clusters two and three are more represented than the activation pattern corresponding to cluster one. Considering the null hypothesis that the distribution of the elements among the clusters is independent of

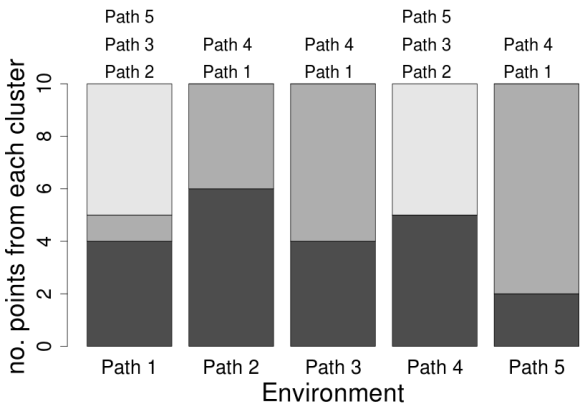



Figure 3-18: Distribution of elements of the 6-dimensional space among the three clusters for each outdoor environment. Black shaded parts of the bars correspond to points belonging to cluster one (see Table 3.6), dark gray shaded parts correspond to points belonging to cluster two and light gray shaded parts correspond to points belonging to cluster three. Annotations above each bar refer to environments where the points distribution on clusters is significantly different (based on Fisher's exact test at 95% confidence level) to the distribution of the environment indicated on the x-axis.

the environments, we carried out the Pearson's chi-squared test of independence. The test gives us a $\tilde{\chi}^2$ value of 56.44, which is higher than the critical value for significance level 0.005 (26.75). This leads us to reject the null-hypothesis and to reach the conclusion that the dynamic color-perception strategy used by the best evolved controller depends on the properties of the environment in which the robot is required to operate. Furthermore, we carried out the Fisher's exact test to statistically verify in which environment the controller displays a different activation pattern of the color parameters. The test indicates that the distribution of elements in Path 1 and in Path 4 is statistically different from the distribution of elements recorded in Path 2, Path 3 and Path 5 (see Figure 3-18). We conclude that, the best evolved controller has different strategies of dynamically activating the color parameters depending on the nature of the outdoor environment, and it adapts its strategy to suit the environment it is operating in.

## 3.7 Conclusion

In this Chapter we tackled the challenge of designing controllers for autonomous vehicles and showed the potential benefits of a design method based on dynamic neural net-

works synthesized by evolutionary computation techniques. By assuming the existence of a colour difference between the road and the non-road areas, we developed a system to design controllers that allow autonomous vehicles to navigate unmarked roads by exploiting this colour difference. We employed artificial neural network based controllers owing to their potential robustness and adaptability to successfully cope with conditions never encountered during training. Visual discrimination tasks requiring colour perception are generally tackled using sensory apparatuses (e.g., a camera) which tend to generate high-dimensional input vectors. With artificial neural networks, high-dimensional input vectors significantly contribute to increase the dimensionality of the parameter search space, with potentially severe consequences on the effectiveness of both supervised and unsupervised training methods to generate successful solutions. We have kept the network parameters search space within reasonable limits by reducing the camera pixels density with a dimensionality reduction process that interfaces the robot camera with the artificial neural network controller. The integrated action-perception approach effectively compensates for the low resolution perceptual system and it allows the robot sensory apparatus to be tuned to the colour characteristics of the environment.

We have described a method that, first, allows generation of robot controllers that can drive autonomous vehicles on unmarked roads by distinguishing the road from the non-road area based on colour differences between the two areas. Second, this method allows the vehicle to autonomously adapt to the variability in colour of the road and non-road areas by generating adaptive mechanisms capable of tuning the robot visual perception system to the characteristics of the environment in which it is required to operate. This is a significant contribution of this study which provides an alternative to the solution described in [91], concerning the challenges faced by vision based navigation systems to cope with the enormous variability of the real world conditions. We have tested our B-controller in a limited set of real world scenes. However, the promising results described in this chapter suggest that this approach can potentially be a valuable option to design control systems for autonomous driverless vehicles required to operate in more challenging conditions.

Another contribution of this work is in illustrating the evolutionary conditions, the functional and structural properties of the network, and the fitness function used to generate

controllers that, when ported to a real robot, proved to be successful and robust enough to deal with complex outdoor scenes. We have extensively observed and analysed the mechanisms underpinning the capability of the B-controller to adjust to the colour characteristics of different environments. We showed that the patterns of activation of the colour parameters $\rho$, $\gamma$ and $\beta$ vary in response to different types of colour changes in the road/non-road areas, both in simulated and real scenarios. We have observed that, although different types of patterns can be generated, the oscillatory patterns are the most frequent ones. We have also run some tests that indicate that the colour parameter activation patterns generated by the networks are absolutely crucial for the functional integrity of the system. In particular, we carried out a variant of *Test 1* (Section 3.4), with the RGB colour model. At each time step of each trial, we substituted the values of $\rho$, $\gamma$ and $\beta$ generated by controller with other values in which the colour parameter associated to the colour component (i.e., R, G, or B) carrying the highest contrast level between road and non-road areas is set to one, and the other colour parameters are set to zero. In this way, the task of the robot should have been facilitated since, in all trials, the vision system was set to maximize the road/non-road contrast. Under this condition, the robot could only successfully terminate 41.50% of the trials, as opposed to 57% when the robot perception system is not altered. This clearly indicates that, although complex activation patterns may appear superfluous, they are indeed playing an invaluable role for the functional integrity of the controller.

# Chapter 4

# Deep Convolutional Neural Network Controller

The work presented in this Chapter aims to address the problem of autonomous driving (especially along ill-defined roads) by using convolutional neural networks to predict the position and width of roads from input images. After the solution proposed in Chapter 3 this is the second approach that explores the development of a controller for autonomous driving across varied road conditions. Initial experiments (see Section 4.1) involved using the principles of evolutionary robotics to synthesize convolutional neural networks with relatively limited dimensions (i.e, number of layers and number of filters per layer). However the sub-par performance of such 'evolved' convolutional networks evaluated in simulated conditions; as well as the recent advances in the use of deep convolutional networks for many computer vision applications discussed in Chapter 2 led us to move towards training larger networks with supervised learning (i.e, back-propagation) using a dataset of annotated road images (see Section 4.2).

We train three different network architectures for images corresponding to six colour models, which are then tested 'off-line' on a road detection task using image sequences not used in training. To benchmark our approach, we compare the performance of our networks with that of a different image processing method that relies on differences in colour distribution between the road and non-road areas of the image (Section 4.1). A trained convolutional network is then used to successfully navigate a Pioneer 3-AT robot on five

distinct test paths constructed in an indoor laboratory. Results from these trials presented in Section 4.4.1 show that the network's road detection outputs can be the basis for guiding the robot in circumstances much different from those represented in the training dataset. After upgrading the computational hardware of the mobile robot with an embedded GPU module (NVIDIA-TX1), we were able to run our CNN models in real time (through CUDA) for outdoor trials in five road/path segments (Section 4.4.2). These outdoor environments were also used to evaluate the neuro-controller with dynamic colour perception (described in the previous Chapter). This provides us metrics with which the road-following/detection solutions developed with two broadly different approaches can be directly compared (see Chapter 5).

## 4.1 Evolving a small Convolutional Neural Netowrk for Road-Following in Virtual Environments

Advancing from the set of experiments involving the dynamic active vision neural network described in Chapter 3, the work presented in this section involves training convolutional neural networks (CNN) using a similar virtual evolutionary learning strategy (opposed to conventional gradient descent techniques). This approach is in part inspired by the set of works described in [64] and [65]. The authors describe experiments wherein they train a CNN using pre-gathered and on-the-fly images from the TORCS driving simulator to act as a feature-extractor for a small recurrent network outputting driving commands. See Section 2.2.3.2 for more details on these works.

Similarly, aiming to benefit from CNNs ability to learn generic feature hierarchies, we use genetic algorithms evaluted in our own virtual driving environment developed for experiments in Chapter 3 as the training scheme. The CNNs could provide a more robust lower-level vision mechanism over the simplistic grid-averaging method for the recurrent network controller used in Chapter 3. Moreover keeping in context the theory of embodiment and motor control being linked to development of visual-perception properties (discussed in Chapter 2.2.3.1), the potential for a solution more advanced than current state of

the art exists by 'evolving' such networks in a virtual simulated environment to be later ported onto a mobile platform. It should be noted that the architecture of the networks used in [64], as well as the experiments presented in this section are quite limited in terms of number of layers and filters in each layer when compared with the scale of the networks that are trained using standard 'off-line' back-propagation techniques. The commonly used AlexNet has 96, 256, 384, 384 and 256 filters in each of its five convolution layers, whereas the network trained using evolutionary algorithms in [64] and [65] besides being limited to a single colour channel input has only 10, 10, 10 and 3 filters in their four layers. Because of the exponential-like cost of weights associated with even incremental increases in higher-level filters, and the resultant increase in search-space for the genetic algoritms, it is more feasible to 'evolve' much smaller CNNs which are similar in scale to the ones employed for hand-writing-recongntion in [73].

### 4.1.1 Method

The architecture of the convolution network generated through evolutionary robotics (or Evo-CNN) as shown in Figure 4-1 is restricted to only two convolution layers. The input consists of a 50×50 greyscale image with R,G and B channels mixed in equal proportions captured from a fixed camera mounted on a Pioneer 3-AT mobile robot. This gets reduced to 6 23×23 feature maps after the first layer of convolution and max-pooling; and 7 3×3 feature maps after the second. It should be noted that a third such round of convolution/max-pooling could not be implemented because of the large computational cost associated with it. Instead the 7×7 pooling operator is applied to reduce the dimmensionality of feature maps generated at the end of the second convolution layer. The four output nodes $\eta_{1,2,3,4}$ are in direct control of the robot's movement on the road. At each time step, the output of the left motor is $M^L = \eta_1 - \eta_2$, and the right motor is $M^R = \eta_3 - \eta_4$, with $M_L, M_R \in [-1, 1]$. The multi-channel convolution operation which takes place at each convolution layer [53] is:

$$y_j = ReLU(\sum k_i j * x_i) \tag{4.1}$$

$$ReLU(x) = max(0,x) \tag{4.2}$$

For the $j^{th}$ filter in a layer $y_j$ is the output corresponding to a particular input patch. $x_i$ is the $i^{th}$ channel of the input and $k^{ij}$ is the corresponding convolution kernel.

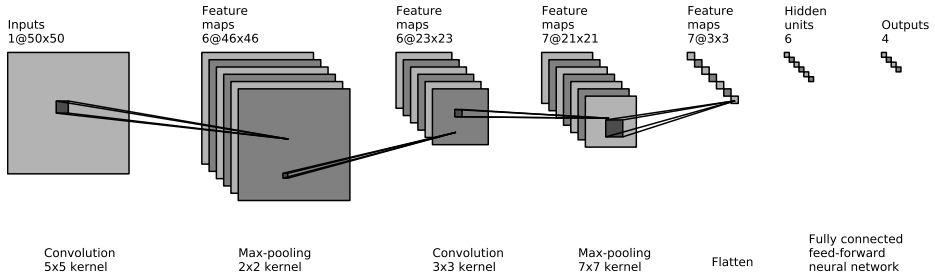



Figure 4-1: Architecture of the Evo-CNN. Sizes of the associated convolution and max-pooling kernels are annotated at the bottom of each relevant layer.

The evolutionary algorithm for synthesizing weights is identical to the one used for the recurrent network controller in the previous Chapter (see Section 3.3.2). The population of $M = 100$ genotypes now represents 930 real values encoding weights of the convolution kernels and fully connected nodes (compared to 243 values earlier). Each individual of the population is evaluated on 12 virtual road environments (shown in Figure 4-2) rendered from a manually chosen set of textures. Because of the absence of any dynamic feedback features in this model, their was no requirement to engineer the textures' colour distributions to impose evolutionary pressures. The principle behind this approach is that the variety presented in the selected texture combinations maybe enough for the networks to 'evolve' sensitivity to a core set of feature-hierarchies that can be generalised to other road-following scenarios. For each environment or 'simulation scene' two road shapes were generated in accordance with the algorithm described in Section 3.3.3.1 resulting in a total of 24 evaluation trials. In each trial $e$, the robot fitness $f_e \in [1.0, 1.5]$ corresponds to the number of road tiles traversed from trial start, and the position in the last traversed tile, in case the robot does not reach the road end within 500 time steps. A trial is terminated earlier

if the robot is detected to have moved off the road. The robot final fitness $F$ is computed as:

$$F = \left(\frac{1}{E}\prod_{e=1}^{E} f_e\right)^E \tag{4.3}$$

$$f_e = 1.0 + \frac{K+H}{22}, \quad H = \frac{Q-V}{Q} \tag{4.4}$$

where $E = 24$ is the total number of trials, $K$ the number of tiles crossed, $Q$ the tile length and $V$ the error vector from mid-point of the tile side closer to the road end to the robot position at the end of the trial. Ten evolutionary runs (corresponding to different seeds for random initialisation) were carried out for a maximum of 1200 generations, with the HPC-Wales cluster being used to evaluate the fitness of each individual in a population in parallel.

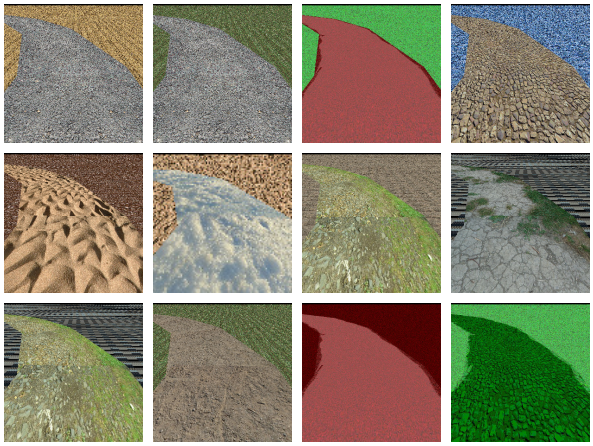


Figure 4-2: Snapshots of the 12 virtual evolution environments used for evolutionary evaluation of the Evo-CNN network (see Figure 4.1).

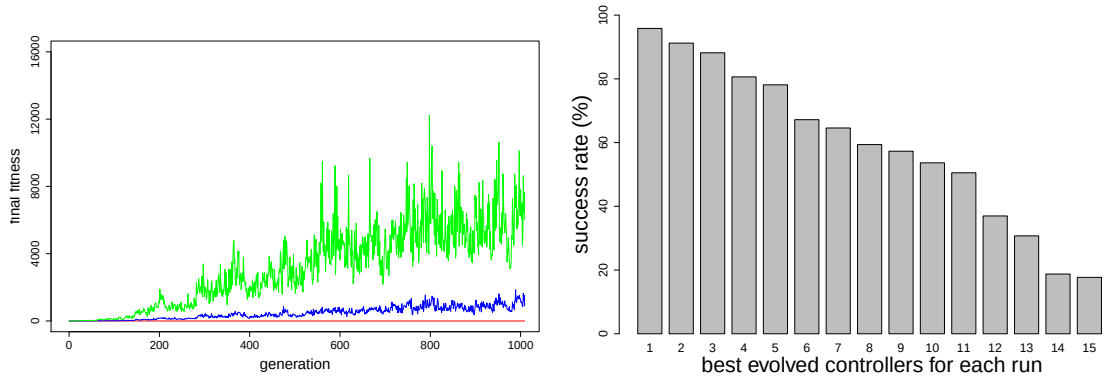
Attempts made to evolve architectures with three convolution layers, greater number of filters in each layer and multiple colour channel inputs were unsuccessful. The reason for this is twofold. Firstly as mentioned earlier the rise in number of parameters associated with increasing network dimensions results in a higher-dimensional search space which means the evolutionary algorithm is prone to get stuck in areas of local optimum in the fitness landscape. Secondly due to the computational cost of propagating values from the input to output layers on even this relatively small CNN model, evolutionary runs for even incrementally larger networks need to extend beyond the allowable time limit for jobs on

the computing cluster available to us.

In order to bench-mark the performance of the Evo-CNN we also evolve a regular 3 layered feed-forward neural network using the same set of 12 virtual road environments shown in Figure 4-2 . Keeping in scale with the CTRNN architecture presented in Chapter 3, the network has 25 input nodes each corresponding to the average of RGB greyscale values of $100\times100$ pixel grids constituting a $500\times500$ raw input image. This is followed by 6 hidden feed-forward nodes and 4 output nodes in control of the robot's motion similar to the Evo-CNN. For the evolutionary algorithm each of the 100 genotypes in a generation represented 176 real values, 174 connection weights and 2 bias values. Evaluating the relative performance of both models, 'evolved' in the same training environment, may help us better understand the advantages/disadvantages of using multi-dimensional and multi-layer convolution over simpler dimmensionality reduction techniques such as pixel-averaging (which require fewer trainable parameters).

### 4.1.2 Results

At the end of the evolutionary training phase, we perform a first post-evaluation test in simulation. The aim of this post-evaluation test is to generate a more accurate estimate of the effectiveness of the most promising solutions in a larger set of operating conditions and help us choose one solution as the 'best' for subsequent tests. We evaluated the best solution (i.e., the best genotype based on fitness) of each generation for the evo-CNN and last 2000 for the feed-forward network. Each solution was re-evaluated on a set of 96 trials based on 8 fixed road shapes in the 12 evolutionary scenes (Figure 4-3). The percentage of success of the best evolved controllers of each evolutionary run, in these 96 post-evaluation trials, is presented in Figure 4-3. The graph shows that 3 out of 10 evolutionary runs managed to produce controllers with a success rate higher than 85%. The most successful Evo-CNN (96% trials) and regular feed-forward controllers (86 % trials) from this stage were chosen for a further round of evaluation on a large set of texture combinations corresponding to the tests (*Test 1*, *Test 2*, *Test 3*, *Test 4*) described in Chapter 3.5. Figure 4-3 shows the maximum, average and minimum fitness per generation of the best evolutionary run.




(a) (b)

Figure 4-3: (a) Fitness graph for best evolutionary run. Green indicates the best, blue the average and red the lowest fitness in a generation. (b) Percentage of success of the best evolved controllers of each evolutionary run, during 96 trials (12 scenes, 8 road shapes). Controllers are ranked from best to worst.

For these tests, 13 possibilities of representing the input image pixel values, using channels from the *RGB*, *HSV*, *YUV* and *lab* colour models were considered. Besides greyscale *RGB* values which was the colour representation during evolution, we tested R,G, B, H, S, U, V, a and b channels individually and HS, UV and ab mixed into single channels with equal ratios of the constituent channels. Channels directly encoding illumination values the in *HSV*, *YUV* and *lab* colour models were not considered. The results of these tests for the Evo-CNN and feed-forward models are summarized in tables 4.1 and 4.2 respectively.

From these tables it can be infered that the simpler 3 layer feed-forward network outperforms the evo-CNN in all four tests, and seems to have developed more generalised visual representations which are better suited for the road-following task. The results of *Test 3*and *Test 4*which involve dealing with a change in road/non-road textures along the course, the relatively poor performance of the Evo-CNN indicates it is less suited for adapting to dynamic changes on the road as well as environments structured differently from those encountered during evolution. For the Evo-CNN there is also greater variability in performance across the colour models it is evaluted in; interestingly performance mostly degrades when colour representations different from RGB greyscale are used (see table 4.1). Unlike the regular feed-forward controller and the recurrent active vision controller from Chapter 3, which benefit from more robust illumination-invariant colour representations, the Evo-CNN seems to rely on extracting regularities that are dependent on the nature of

Table 4.1: Performance of the colour models with Evo-CNN, for *Test 1*, *Test 2*, *Test 3*, *Test 4*, *Test 5*. For each test, the colour models are ranked in descending order of performance.

| *Test 1* | | *Test 2* | | *Test 3* | | *Test 4* | |
|---|---|---|---|---|---|---|---|
| **Colour channels** | **Success rate (%)** | **Colour channels** | **Success rate (%)** | **Colour channels** | **Success rate (%)** | **Colour channels** | **Success rate (%)** |
| R | 53.8 | RGB | 40.0 | R | 41.7 | RGB | 37.3 |
| RGB | 52.7 | R | 40.7 | RGB | 41.4 | HS | 25.9 |
| G | 51.9 | B | 39.7 | B | 39.9 | ab | 10.4 |
| B | 50.6 | G | 39.1 | G | 39.4 | UV | 8.1 |
| HS | 39.4 | S | 29.1 | S | 29.2 | R | 0 |
| S | 38.1 | HSS | 22.6 | HS | 21.7 | G | 0 |
| H | 27.2 | H | 14.2 | H | 13.6 | B | 0 |
| ab | 25.5 | ab | 12.7 | ab | 11.6 | H | 0 |
| UV | 23.2 | V | 10.9 | V | 10.5 | S | 0 |
| U | 22.2 | U | 9.5 | U | 7.8 | U | 0 |
| V | 21.9 | a | 9.0 | a | 7.1 | V | 0 |
| a | 20.2 | b | 8.6 | V | 6.9 | a | 0 |
| b | 19.1 | UV | 7.1 | b | 5.8 | b | 0 |

colour distributions encountered during evolution, i.e. the learning phase. A similar observation is also made during later experiments presented in Section 4.2 involving training much larger CNNs for a regression problem (predicting road boundaries) using the more standard-backpropagation technique. CNNs could only exhibit error rates within acceptable margins when tested with image-sets using the colour scheme used during training.

Across all four tests, results from table 3.12 (Chapter 3) point to the superiority of the previously implemented active vision neuro-controller with its dynamic colour perception strategy over the Evo-CNN and regular feed-forward architectures. Despite its sub-par performance, we were able to demonstrate that a small convolutional neural network, with relatively few trainable kernels, can be 'evolved' to navigate a robot on simulated roads of varying shapes and colour properties. However computational costs associated with the evolutionary training process and no clear performance advantages when bench-marked with a much more simplistic neurocontroller architecture meant a different approach was required for generating convolutional networks capable of controlling robots in noisy outdoor visual scenes. This experiment validates the feasability of online evolution for generating CNN based road-following controllers; an approach also presented in [65]. Inter-

estingly somewhat mirroring our findings, driving results from a set of test courses in the TORCS racing simulator presented in [65], show a standard recurrent neural network receiving direct pixel values from a 63×63 image as its input vector performed better than the convolutional architecture proposed by the authors. We thus shift our focus to training much 'larger' convolution neural networks through the more commonly employed back-propagation/gradient-descent approach.

Table 4.2: Performance of the colour models with a benchmark 3 layered feed-forward network, for *Test 1*, *Test 2*, *Test 3*, *Test 4*, *Test 5*. For each test, the colour models are ranked in descending order of performance.

| *Test 1* | | *Test 2* | | *Test 3* | | *Test 4* | |
|---|---|---|---|---|---|---|---|
| **Colour channels** | **Success rate (%)** | **Colour channels** | **Success rate (%)** | **Colour channels** | **Success rate (%)** | **Colour channels** | **Success rate (%)** |
| H | 59.2 | H | 57.0 | H | 48.5 | H | 42.3 |
| HS | 54.9 | S | 51.0 | S | 45.2 | b | 38.0 |
| S | 54.7 | b | 49.9 | HS | 43.2 | S | 37.8 |
| b | 49.6 | HS | 48.9 | b | 41.7 | V | 35.2 |
| V | 47.6 | UV | 48.2 | UV | 39.0 | a | 35.1 |
| a | 46.2 | a | 47.5 | a | 38.9 | HS | 34.6 |
| UV | 45.6 | V | 45.9 | U | 38.8 | ab | 33.4 |
| ab | 45.2 | U | 45.5 | V | 37.4 | UV | 32.6 |
| U | 43.4 | ab | 43.1 | ab | 36.7 | U | 30.4 |
| B | 25.9 | B | 34.4 | B | 29.1 | B | 19.9 |
| RGB | 25.2 | RGB | 31.2 | RGB | 26.7 | RGB | 16.7 |
| R | 24.1 | G | 27.3 | G | 23.1 | R | 16.0 |
| G | 23.5 | R | 24.7 | R | 20.5 | G | 15.2 |

## 4.2 Road Detection using Deep Convolutional Neural Networks

Training 'deep' convolutional networks to directly control the motion of a mobile platform whilst desirable, can be difficult to achieve. Experimental results from the previous section suggest that it is impractical to consider the embodied evolutionary robotics approach from Chapter 3 as this can be applied only for neural models with a limited number of trainable

parameters. An alternative to this can be to meticulously generate road image-sequences, with labelled snapshots corresponding to a set of possible control decisions (e.g. turn degrees) at every point. A convolutional network can then be trained through backpropagation to predict these control outputs that further decompose to motor commands. Similar techniques were used to generate datasets for the ALVINN project [94] which involved training a neural network to predict motion control commands given an input camera image. However as discussed in Chapter 2.2.1 despite dedicating a substantial part of the project to this aspect, the relatively small number of 1200 data points thus generated can be representative of only a small fraction of actual road conditions/scenarios that an autonomous vehicle may encounter. As such convolutional neural networks with larger dimensions are powerful architectures that are prone to over-specialise/over-fit in cases of limited training examples. To generate larger datasets mapping raw input images to a practical range of possible navigational commands involves hours of manual annotation that was not considered feasible. We therefore view the application of convolutional networks in this problem domain to be more viable as 'road-detectors'; rather than integrated sensorimotor controllers.

For the work presented in this section we train and evaluate convolutional neural networks of three different architectures on 20 datasets of road image sequences representative of a variety of environments (urban and rural). We train the CNNs on different colour models to explore whether differences in colour representation influence the network performance. A detailed description of the colour models used (*RGB*, *HSV*, *YUV*, *YCbCr*, *lab*, *HS*, *UV*, *CbCr*, *ab*, and *CbCra*) can be found in Appendix B. *HS*, *UV*, *CbCr*, *ab*, and *CbCra* (a hybrid model derived from *YCbCr* and *lab*) are included to test the effectiveness of discarding luminance information from the input image. We also evaluate the performance of the 'adaptive statistical colour' based method described in [91] (henceforth also referred to as ASC) for these datasets, providing a bench-mark for how well the networks perform. The results of our study show that the best performing networks match the ASC methods detection accuracy for majority of the datasets and even outperform it for two environments. To further validate our method, we ported our trained networks onto a Pioneer 3-AT robot, and tested the system in five test paths constructed indoors and five outdoor road segments (see Section 4.4).

### 4.2.1 Road-Model

Drawing inspiration from the method described in [91], we use a trapezoidal model with two changeable parameters to provide a best-fit to the road in the input image. The convolution neural network generates two parameters, the position $(x)$ and the width $(w)$ of the trapezoid delimiting the road area. It is acknowledged that varying other parameters such as $\theta$ and $h$ (see Figure 4-4) could have provided a closer fit between the trapezoid and the road. However, the addition of further parameters inevitably increases the complexity of task for the network, and it clearly generates overheads to the process of annotating the images for the network training. Indeed the position $x$ on its own provides enough information for a robot to stay on the road. The width $(w)$ parameter is required if the speed of the robot needs to be varied and is useful for more complex control procedures resulting in smoother less oscillatory motion.

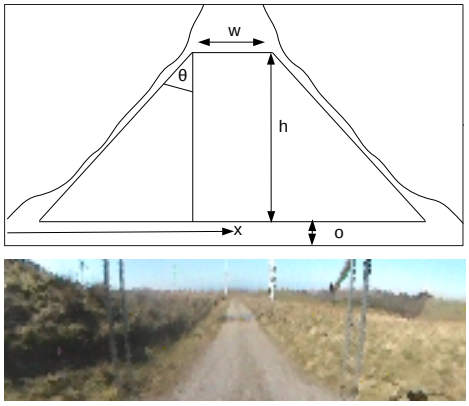



Figure 4-4: Parameters of the trapezoidal road model and the projection of the model on an image of a road.

### 4.2.2 Network-Architectures

Besides exploring colour representation, we also investigate the effect of neural model complexity on the road detection accuracy. Do deeper and more complex neural network architectures give us better detection results? For this purpose we train and evaluate three different sized convolutional neural network architectures and a simple 4-layered feed-forward network. The feed-forward network receives a $50\times50$ 3 channel image which is

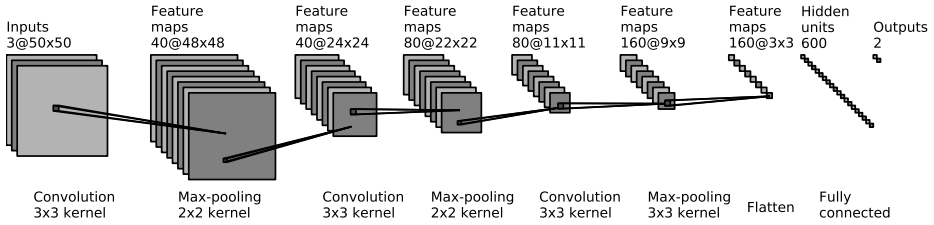



Figure 4-5: Architecture of the LCNN (see Section 4.2.2). Sizes of the associated convolution and max-pooling kernels are annotated at the bottom of each relevant layer.

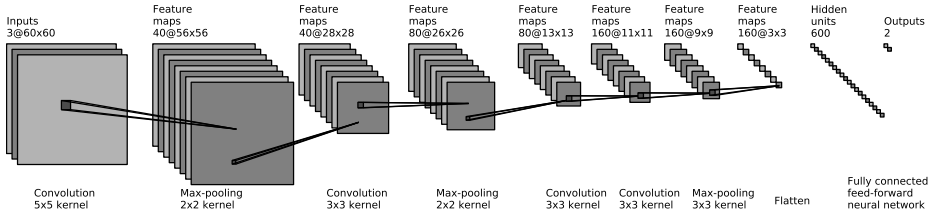



Figure 4-6: Architecture of the MCNN (see Section 4.2.2). Sizes of the associated convolution and max-pooling kernels are annotated at the bottom of each relevant layer.

flattened to form an input vector of 7500 input neurons. It has 2 hidden layers with 1000 and 600 neurons respectively and two nodes coding for the parameters $x$ and $w$, which define the position and width respectively of the road model (see Section 4.2.1). The values of $x$ and $w$ are then mapped to the range of 0-180 pixels. The results of this network were very poor with average position errors greater than 20 pixels across all datasets. These results (not discussed further in this work) establish the fact that a more complex model than a regular feed-forward neural network is required for this task. The 'smallest' CNN architecture, which we shall refer to as the Light Convolutional Neural Network or LCNN

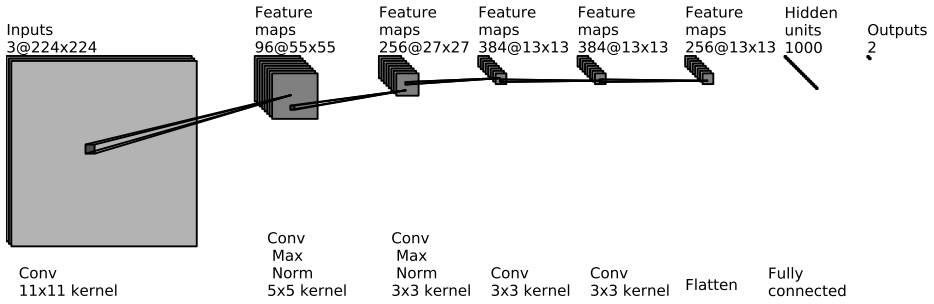



Figure 4-7: Architecture of the modified AlexNet (see Section 4.2.2). Conv, Max, Norm refer to convolution, max-pooling and local contrast normalization operations respectively (see [68]). The sizes of the associated convolution kernels for each layer are annotated at the bottom of the image.

can be seen in Figure 4-5. Equations 4.1 and 4.2 describe the multi-channel convolution operation which takes place at each convolution layer [53]. The rectified linear activation unit (equation 4.2) is better suited for convolutional neural networks and provides a more biologically plausible modelling of neural activity over the traditional sigmoid activation function. See [40] for a detailed justicification of the use of rectified activation units for deep multi-layer architectures. For the $j^{th}$ filter in a layer $y_j$ is the output corresponding to a particular input patch. $x_i$ is the $i^{th}$ channel of the input and $k^{ij}$ is the corresponding convolution kernel. The network follows the conventional expanding shape approach used for most architectures by having 40, 80 and 160 filters in its first, second and third convolution layers. It receives a $50\times50$ image as its input. Depending on which colour model is being used the image is split up into its constituent channels and these are individually fed into the filters for the first layer. The network activations are then propagated to the output layer, made of two output nodes predicting the values of $x$ and $w$. Weights are initialized randomly and updated using a variant of standard mini-batch gradient descent called rmsprop [27]. To prevent over-fitting of these complex models, dropout noise (see [106]) of 0.2 and 0.5 is used at the convolution and fully connected layers respectively.

The second architecture, MCNN (Medium Convolutional Neural Network) shown in Figure 4-6 is very similar to LCNN with 4 convolutional layers (instead of 3) and a reduced learning rate ($\eta$) of 0.0005. The additional layer of convolution is incorporated to gauge the effectiveness of another level of representation in the feature hierarchy. Networks of both architectures are trained for a maximum of 100 epochs, using Theano [113] running on the HPC-Wales GPU cluster. This enables us to generate and evaluate multiple networks (corresponding to different colour model, network architecture combinations) in parallel, a process which would have otherwise have taken much longer if a single GPU host were to be used.

Besides the LCNN and MCNN, we also train a slightly modified version of the AlexNet architecture described in [68]. As shown in Figure 4-7 this is a much larger and deeper network with as many as 5 convolution layers, compared to the LCNN architecture described above. Similar to the LCNN it has 2 output nodes predicting the values of $x$ and $w$ and receives a 3 channel image as its input. Refer to [68] for a more detailed overview of

this architecture. The original unmodified architecture has 2 fully connected layers (with 4096 neurons each) after the fifth convolution layer. We take a network with this original architecture trained on the ImageNet dataset and remove the fully connected hidden and output layers, replacing them with a randomly initialized hidden and 2 node output layer. The entire model (end-to-end) is then re-trained for 30 batch iterations (lesser training time because of pre-initialized weights) with GPU acceleration using Caffe (see [56]), using the step learning policy (base lr = 0.0001). While the LCNN and MCNN architectures are trained with all 10 colour models/combinations listed earlier in this section, the AlexNet is trained with three channel colour models (*RGB*, *HSV*, *YUV*, *YCbCr*, *lab* and *CbCra*). This is because the AlexNet architecture inherently receives a three channel image as its input and since we are interested in the effects of fine-tuning such a network already trained on a larger general image dataset, only the 6 above mentioned models are considered.

### 4.2.3 Datasets

As mentioned before we want to train our networks to be able to detect roads in any environment irrespective of colour, lighting, geometry etc. Networks should also be flexible enough to work for images captured on any platform and camera configuration. For this reason a total of 20 datasets corresponding to road-sequences in varied environments and captured from 6 different sources (see table 4.2.2) are used to train and evaluate our network. Each image in these datasets is manually annotated with the position and width parameters of the road model described in section 4.2.1. We divide these datasets into two categories, keeping some for training/validation and others only for testing. The datasets for training and validation are split into two halves, with the first half used for training and latter for validation. An exception to this is Lakeside where a much larger portion of the frames are used for validation. As this is a much larger image sequence, splitting the dataset into equal halves could mean too many images from this environment being included in the overall training set. The performance of a network in the validation sets points to its ability to deal with new image sequences in environments it has already been trained for, whereas in the test sets it deals with images from environments previously not encountered.

| Name | No. Images | Platform | Purpose |
|---|---|---|---|
| CMU | 296 | NA | Train/Valid |
| W.Farm | 2998 | IDRIS Rover | Train/Valid |
| K39 | 188 | KITTI | Train/Valid |
| K23 | 242 | KITTI | Train/Valid |
| Footpath | 1551 | Pioneer 3-AT | Train/Valid |
| K34 | 255 | KITTI | Test |
| K56 | 154 | KITTI | Train/Valid |
| K59 | 100 | KITTI | Test |
| K86 | 296 | KITTI | Train/Valid |
| K87 | 296 | KITTI | Test |
| K93 | 126 | KITTI | Train/Valid |
| Lakeside | 8024 | IDRIS Rover | Train/Valid |
| Llan. | 541 | IDRIS Rover | Test |
| Misc | 128 | NA | Train/Valid |
| Tiled | 1380 | GOPRO Stick | Test |
| Rain | 838 | GOPRO Stick | Test |
| Rugged | 2760 | GOPRO Stick | Train/Valid |
| Shadows | 150 | IDRIS Rover | Test |
| Steep | 2033 | GOPRO Stick | Test |
| R.Track | 1802 | Pioneer 3-AT | Train/Valid |

Table 4.3: Table detailing the size, camera platform of each dataset and whether they were used for training/validation or only testing. Pioneer 3-AT and IDRIS Rover are mobile robots available with the department and the images were captured while manually driving the robots along these roads. GOPRO Stick refers to images captured by a GoPro Hero4 camera (https://gopro.com). KITTI refers raw images downloaded (and later resized/annotated) from the KITTI Vision Benchmark (see [38]). CMU and MISC are composed of images gathered from an online image repository and the photo sharing website 'Flickr' respectively.

**CMU**

This dataset consists of 296 images of a mobile-robot looking down upon roads in various environments, obatined from the CMU (Carnegie Mellon University) computer vision repository at `http://www.cs.cmu.edu/afs/cs/project/vision/vasc/idb/www/html_permanent/`. Along with MISC this is not a sequential dataset, but rather contains snapshots of roads of different surfaces (tarmac, dirt, potholed etc.) and lighting, weather conditions such as extreme sunlight, shadows, snow. Besides for a few images the white balance settings of the camera caused a reddish tint which added another complexity

to the detection.

Figure 4-8: Frames from dataset CMU.

**Wind Farm**

This dataset contains 2998 frames captured from an omni-directional camera mounted on the IRDIS rover robot available with the Computer Science Department at Aberystwyth University. The challenges of this dataset include low contrast between the road and the surroundings with no crisp edge, surface discoloration due to wet patches, and changing viewpoint due to changes in the on-road position. (This description has been included from [91], where the dataset was originally presented).

Figure 4-9: Frames from dataset Wind Farm.

**K39**

This dataset contains 188 frames of a mobile robot moving through an urban road with parked vehicles and buildings on either side. These images were obtained from the KITTI dataset (see [38]) and then transformed and annotated by us. The major challenges are shadows from buildings and trees and the presence of static/moving vehicles of various sizes in the road scene.

**K23**

This dataset contains 242 frames of a mobile robot moving through a street with the borders being marked by trees, grassy patches and parked vehicles on either side. These images

Figure 4-10: Frames from dataset K39.

were obtained from the KITTI dataset (see [38]) and then transformed and annotated by us. There are drastic changes in illumination along the road alternating between shadows and regions of bright sunlight.

Figure 4-11: Frames from dataset K23.

**Footpath**

The road in this dataset is a marked tarmac surface that contrasts well against the mostly green (grass) and blue (running track) surroundings. This dataset comprises 1,551 images captured from an omni-directional camera mounted on a Pioneer 3-AT mobile robot available with the Computer Science Department at Aberystwyth University. The challenges of this dataset include a crossroad, a widening of the road, and an obstacle. Also, the tarmac is covered in moss on the left-hand side at the beginning of the dataset.

Figure 4-12: Frames from dataset Footpath.

**K34**

This dataset contains 255 frames of a mobile robot moving through a sub-urban street. These images were obtained from the KITTI dataset (see [38]) and then transformed and

annotated by us. There is a considerable environmental variability within this dataset as the sequence begins with the green fields on both sides, moving on to a residential area with houses and parked cars along the road boundaries. Other challenges include sharp turns, varying illumination and moving vehicles.

Figure 4-13: Frames from dataset K34.

**K56**

This dataset contains 154 images in an urban environment originally obtained from the KITTI dataset (see [38]). The images show well a demarcated (white lines) two lane highway road. The detection algorithm/method is requried to only consider the lane the recording platform was driving on (i.e the right lane). Apart from this other challenges include, shadows, variations in lane-markings and presence of cars in both lanes

Figure 4-14: Frames from dataset K56.

**K59**

This dataset contains 100 images in an urban environment originally obtained from the KITTI dataset (see [38]). The dataset involves travelling along well demarcated road with kerbs, parked cars, pavements and other road-lanes on the sides. Challenges involve presence of traffic, shadows, an intersection and very low contrast between the road-lane and pavement.

Figure 4-15: Frames from dataset K59.

**K86**

This dataset contains 296 frames in a uphill/inclining sub-urban street originally obtained from the KITTI dataset (see [38]). This is another dataset with a greater degree of environmental variability, especially with regards to the non-road areas which include buildings, sheds, trees, hedges and parked cars. Besides this some of the other complexities of this image sequence are changing global illumination, irregular shadows and a narrowing of the road mid-way.

Figure 4-16: Frames from dataset K86.

**K87**

This dataset contains 296 frames in a sub-urban road originally obtained from the KITTI dataset (see [38]), where the mobile vehicle moves from a regular black tarmac road surrounded primarily by hedges, trees and fences to a wider road predominantly surrounded by houses. Besides shadows and sharp changes in overall illumination, there is variability in the surface of the road as well, especially towards the end of the image sequence where repairs seem to have caused patches that are coloured differently to the rest of the road.

Figure 4-17: Frames from dataset K87.

### K93

This dataset contains 126 frames in an urban street originally obtained from the KITTI dataset (see [38]). There is no clear demarcating road edge, with parked cars and buildings on either side providing cues for delineation. Moreover there is a sharp right turn that needs to be navigated along the course. Most of the road is entirely covered in shadows, with occasional stretches of bright sunlight.

Figure 4-18: Frames from dataset K93.

### Lakeside

This dataset contains 8024 frames (with only 2006 used for training)captured from an omni-directional camera mounted on the IRDIS rover robot available with the Computer Science Department at Aberystwyth University. The road in this dataset is made of various materials ranging from loose gray gravel to brown mud, and it presents dry and wet patches with puddles in places. The road is delimited by grass, but the boundary road grass is not always obvious. (This description has been included from [91], where the dataset was originally presented).

Figure 4-19: Frames from dataset Lakeside.

### Llanbadarn

This dataset contains 541 frames captured from an omni-directional camera mounted on the IRDIS rover robot available with the Computer Science Department at Aberystwyth University. The road consists of a mostly well-marked tarmac surface with mostly grass/shrubs

on the sides. The challenges of this dataset include noise (a moving diagonal pattern in the original omnidirectional images resulting in hyperbolic lines forming on the unwrapped panoramic images due to noise in the camera), general lack of color contrast (due to an overcast sky), raised pedestrian crossings, a T-junction, and a sharp turn involving a brief change in the road surface. (This description has been included from [91], where the dataset was originally presented).

Figure 4-20: Frames from dataset Llanbadarn.

### Misc

This dataset contains 128 independent road images, rather than being continuous sequential frames (similar to CMU). These images were gathered from the popular photo-sharing website Flickr (`https://www.flickr.com/`) through an automated script which downloads images that were tagged with keywords such as 'road','street','path' etc. The script only downloads images which had the suitable creative commons license (CC) for academic usage. This dataset is incorporated to increase overall training-set variability and evaluate the effectiveness of our networks in detecting roads irrespective of the camera angle/configuration and environmental conditions in par with human vision.

Figure 4-21: Frames from dataset Misc.

### Tiled

This dataset contains 1380 frames captured from a GOPRO Hero4 camera (http://gopro.com), on a path made up on tiled stones on the campus of Aberystwyth University, filmed during

a spell of rain. The non-road surface on the side varies from grassy patches to tall hedges. Segments of this path were used for outdoor road trials using detection outputs from trained convolutional neural networks presented in Section 4.4.2. We also carried outdoor trials with the controller developed in Chapter 2 in these test environments.

Figure 4-22: Frames from dataset Tiled.

### Rain

This dataset contains 838 frames captured from a GOPRO Hero4 camera (http://gopro.com). The smooth tarmac road in this dataset was filmed immediately after a spell of rain had cleared up resulting in bright spots of reflection on the surface. The non-road surfaces on the sides are quite varied and include wooden barricades, grassy patches, shrubs, benches, thrash bins and a pedestrian crossing the robots path while walking along the road edges.

Figure 4-23: Frames from dataset Rain.

### Rugged

This dataset contains 2760 frames captured from a GOPRO Hero4 camera (http://gopro.com). The road consists of a hilly path made of primarily dirt and gravel (filmed after a spell of rain). Challenges include low contrast (due to overcast conditions), puddles, extremely delineated edges at some sequences and occasional sharp turns. There are also a few sudden changes in colour properties with the appearance of fences and barricades on the sides.

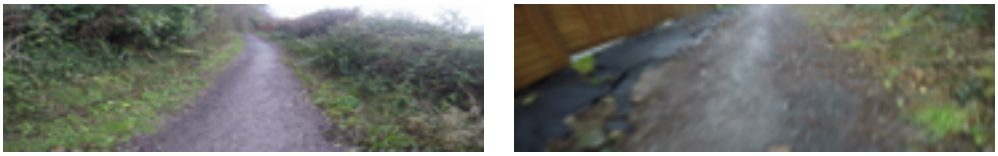


Figure 4-24: Frames from dataset Rugged.

**Shadows**

The road in this dataset is a well-marked tarmac surface that contrasts against the mostly green surroundings. This dataset comprises 150 images and is geographically a subset of the Llannbadarn dataset, not including any of the intersections and driven in the opposite direction. The challenges of this dataset involve saturation and therefore high and low (color) contrasts, coupled with multiple shadow configurations on the road. Note that on that dataset, the ground truth covers the whole road width, therefore including both bright (almost saturated) and shaded areas, while the initial model only contained the bright area (because of initial positioning of the robot). (This description has been included from [91], where the dataset was originally presented).

Figure 4-25: Frames from dataset Shadows.

**Steep**

This dataset contains 2033 frames captured from a GOPRO Hero4 camera (http://gopro.com). The road is a steep tarmac path with leaves and mud covering sections of it. Indeed one of the most challenging aspects of this dataset is the leaves on either side of the road that make it difficult to ascertain where the actual road boundaries are. Other challenges include changes in non-road surfaces with appearance of fences on either side and occasional widening of the road.

Figure 4-26: Frames from dataset Steep.

**Running Track**

This dataset contains 1802 frames captured from an omni-directional camera mounted on a Pioneer 3-AT mobile robot available with the Computer Science Department at Aberystwyth University. The running track is a highly visible manufactured surface accompanied by well-defined edge markings. This blue surface contrasts well against the white edges and central line as well as the grassy surroundings. However, the two adjacent running tracks are both some variation of blue, the robot driving on the lighter of the two tracks. The challenges of the dataset include changes in surface color caused by leaves, shadows cast by nearby trees, occasional crossing white and yellow lines, and an intersection. (This description has been included from [91], where the dataset was originally presented).

Figure 4-27: Frames from dataset Running Track.

## 4.3 Offline-Detection Results and Analysis

After training we select the best network from each colour model of each architecture for off-line evaluation. For this work networks are selected solely based on their accuracy in predicting the position $(x)$ parameter, as position information is more important for navigation than width $(w)$. We look for networks that can perform to a reliable degree of accuracy across all datasets rather than networks which minimize error for a few datasets but fail in the rest. For each colour model we set this average error threshold at 9 pixels per dataset. If no networks are found matching this criteria for one or more datasets we

Table 4.4: Median and Standard Deviation of position error (in pixels) of best LCNN for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

| Dataset | RGB | | HSV | | HS | | lab | | ab | | YUV | | UV | | YCbCr | | CbCr | | CbCra | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std |
| R.Track | -2.1 | 4.4 | 4.4 | 7.6 | **0.0** | **0.0** | -1.3 | 1.6 | 0.3 | 1.4 | -3.9 | 1.6 | -2.9 | 1.3 | -5.9 | 1.7 | 0.2 | 2.0 | 0.0 | 1.2 |
| Llan. | 20.4 | 11.6 | 11.0 | 8.9 | -4.8 | 8.7 | 2.2 | 5.2 | **1.2** | **5.2** | -1.9 | 5.6 | -1.4 | 5.4 | -3.2 | 5.9 | -1.2 | 5.6 | -2.8 | 5.4 |
| W.Farm | -1.8 | 10.6 | 5.1 | 9.0 | 0.7 | 9.3 | -1.9 | 7.2 | -2.1 | 7.0 | -1.1 | 7.8 | -1.8 | 7.3 | -3.6 | 7.7 | **-0.1** | **7.4** | -5.3 | 6.9 |
| Rugged | -13.8 | 10.5 | 4.2 | 7.3 | 6.6 | 9.5 | -2.2 | 9.1 | -1.8 | 9.0 | -5.2 | 9.1 | **-1.7** | **8.9** | -5.9 | 9.0 | -3.8 | 8.8 | -3.1 | 8.9 |
| Rain | -7.2 | 11.6 | -4.0 | 7.4 | -14.3 | 10.6 | -3.7 | 10.4 | -3.3 | 10.4 | **-2.1** | **12.3** | -4.2 | 11.0 | -3.9 | 12.4 | -3.7 | 9.5 | -5.7 | 10.9 |
| Footpath | 7.8 | 10.7 | 3.3 | 9.4 | **-0.2** | **8.3** | -2.1 | 8.2 | -0.7 | 8.0 | -9.1 | 7.6 | 0.3 | 7.3 | -12.2 | 7.7 | -1.8 | 7.8 | -2.1 | 7.4 |
| MISC | -11.5 | 35.9 | 7.1 | 28.9 | -2.3 | 31.6 | 3.9 | 28.7 | 2.4 | 28.1 | -2.4 | 26.3 | **-1.6** | **26.5** | -6.4 | 26.9 | -3.5 | 26.8 | -3.4 | 26.3 |
| CMU | -19.7 | 35.8 | **0.0** | **27.8** | 4.4 | 32.7 | -9.9 | 13.3 | -8.3 | 12.3 | -11.7 | 15.3 | -7.5 | 19.7 | -12.0 | 14.9 | -8.8 | 17.7 | -8.9 | 18.5 |
| K87 | -2.1 | 15.9 | -8.5 | 10.7 | -6.7 | 15.3 | 1.1 | 7.9 | 0.3 | 8.0 | 1.1 | 8.1 | **-0.2** | **8.2** | -1.6 | 8.1 | -2.8 | 9.0 | -1.9 | 8.0 |
| K59 | -6.3 | 6.0 | 4.3 | 3.4 | 8.9 | 5.1 | -1.9 | 1.1 | -0.8 | 1.0 | -0.9 | 3.1 | 1.8 | 1.7 | -1.9 | 2.3 | **-0.4** | **1.6** | -0.6 | 1.4 |
| K34 | **1.1** | **22.5** | 8.7 | 19.2 | 11.7 | 21.8 | 3.5 | 17.6 | 4.1 | 18.1 | 8.6 | 18.4 | 7.1 | 19.5 | 6.8 | 18.6 | 5.6 | 19.1 | 4.4 | 19.1 |
| Steep | -13.0 | 6.3 | **1.9** | **6.0** | 11.7 | 21.8 | -6.4 | 5.4 | -5.3 | 5.5 | -4.5 | 5.9 | -4.2 | 5.8 | -5.1 | 6.0 | -5.6 | 5.8 | -5.8 | 5.7 |
| Lakeside | -2.3 | 9.3 | 13.2 | 5.0 | **-0.7** | **4.0** | -1.6 | 2.3 | -2.7 | 2.2 | -5.4 | 3.5 | -3.6 | 1.9 | -8.8 | 3.3 | -1.7 | 2.3 | -5.8 | 1.9 |
| Tiled | -1.3 | 9.3 | 11.1 | 6.5 | **-1.2** | **9.1** | -3.4 | 8.8 | -5.5 | 8.4 | -4.0 | 9.6 | 6.1 | 8.8 | -5.7 | 10.2 | -6.0 | 8.2 | -7.7 | 8.8 |
| K23 | 8.1 | 11.5 | 5.1 | 7.3 | -1.2 | 9.1 | 4.1 | 3.5 | 4.9 | 3.3 | 6.6 | 5.1 | 4.4 | 3.7 | 4.1 | 4.6 | **-1.1** | **4.0** | 2.5 | 3.4 |
| K39 | 4.7 | 15.8 | 12.1 | 12.7 | 16.9 | 14.0 | **0.3** | **11.3** | 1.9 | 11.2 | 6.4 | 11.7 | 10.4 | 10.9 | 7.4 | 12.9 | 6.1 | 10.4 | 6.1 | 10.8 |
| K56 | 5.9 | 6.5 | 0.5 | 3.9 | -2.2 | 7.0 | -3.2 | 1.0 | -2.4 | 1.0 | 6.7 | 4.2 | **-0.8** | **2.0** | 3.5 | 4.4 | -1.6 | 1.4 | -2.8 | 1.8 |
| K86 | -1.2 | 13.0 | -7.7 | 11.3 | -15.2 | 20.2 | 4.8 | 5.8 | 4.4 | 5.5 | **-0.6** | **8.0** | -1.5 | 5.9 | -2.6 | 7.1 | -4.3 | 5.5 | -1.8 | 5.8 |
| K93 | -14.1 | 21.1 | **-3.6** | **20.1** | -5.2 | 18.6 | -14.1 | 15.8 | -12.6 | 16.0 | -7.3 | 19.4 | -9.2 | 16.4 | -10.2 | 18.1 | -13.5 | 16.6 | -11.5 | 16.2 |
| Shadows | 16.8 | 11.6 | -3.9 | 4.8 | -4.3 | 8.5 | 12.2 | 2.7 | 11.7 | 2.7 | -0.8 | 5.3 | 9.3 | 2.6 | **0.4** | **4.6** | 8.4 | 3.1 | 7.5 | 2.7 |

Table 4.5: Median and Standard Deviation of position error (in pixels) of best MCNN for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

| Dataset | RGB | | HSV | | HS | | lab | | ab | | YUV | | UV | | YCbCr | | CbCr | | CbCra | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std |
| R.Track | -4.9 | 6.4 | 15.2 | 8.5 | -13.6 | 4.1 | 0.8 | 1.5 | 2.3 | 1.4 | 0.9 | 1.2 | **0.0** | **1.3** | -2.0 | 1.6 | 3.7 | 1.3 | 3.1 | 1.7 |
| Llan. | 18.2 | 12.6 | 3.4 | 9.9 | -11.4 | 9.0 | 4.5 | 5.2 | 3.7 | 5.1 | 0.5 | 5.3 | **-1.0** | **5.3** | 5.1 | 6.1 | 3.5 | 5.3 | 3.0 | 2.9 |
| W.Farm | 2.3 | 11.2 | -1.6 | 12.4 | -15.1 | 13.2 | **0.4** | **7.2** | 0.5 | 7.1 | -4.6 | 6.7 | -6.2 | 6.6 | -0.4 | 7.3 | -2.2 | 6.6 | -1.7 | 4.5 |
| Rugged | -12.7 | 10.6 | -4.4 | 7.7 | 18.0 | 10.9 | -0.8 | 9.0 | **0.2** | **9.0** | -1.0 | 8.9 | -1.6 | 8.9 | -4.0 | 8.8 | 2.4 | 8.8 | 2.0 | 9.3 |
| Rain | -6.7 | 11.6 | -14.8 | 9.4 | -1.2 | 13.9 | -2.8 | 10.3 | -1.5 | 10.4 | -4.2 | 10.6 | -4.5 | 10.7 | -5.1 | 10.5 | **-0.7** | **10.6** | -4.9 | 12.1 |
| Footpath | -10.7 | 10.8 | 2.8 | 11.5 | -9.3 | 7.6 | **0.0** | **8.1** | 1.4 | 8.0 | -0.4 | 7.5 | -1.7 | 7.5 | -4.1 | 7.5 | 2.4 | 7.5 | 3.2 | 7.0 |
| MISC | -7.6 | 35.3 | -20.5 | 36.4 | -10.2 | 34.3 | 4.5 | 28.9 | 5.9 | 28.3 | -3.5 | 27.1 | -4.8 | 26.5 | -3.5 | 27.7 | **-0.5** | **26.8** | 3.2 | 7.0 |
| CMU | -18.5 | 35.2 | -16.8 | 34.4 | 16.2 | 42.9 | -7.3 | 12.8 | -7.1 | 12.1 | -7.7 | 13.6 | -8.4 | 16.0 | -11.3 | 17.2 | **-4.0** | **15.2** | 6.4 | 15.3 |
| K87 | 4.4 | 14.3 | -16.6 | 13.7 | -7.9 | 17.2 | 3.0 | 7.9 | 2.6 | 7.9 | **0.0** | **7.6** | -0.8 | 7.7 | 1.3 | 7.7 | 3.1 | 7.7 | 5.4 | 20.9 |
| K59 | -6.7 | 6.3 | -10.5 | 5.8 | 5.1 | 4.2 | **0.5** | **1.0** | 1.2 | 1.0 | 0.6 | 1.0 | -0.6 | 1.1 | -0.9 | 1.7 | 3.4 | 1.0 | 1.1 | 3.8 |
| K34 | 17.8 | 25.6 | 25.9 | 16.2 | 22.3 | 15.7 | 4.9 | 10.7 | 9.6 | 10.5 | 11.4 | 12.6 | 19.9 | 14.9 | **4.3** | **11.9** | 12.6 | 19.1 | 3.6 | 14.3 |
| Steep | 11.3 | 21.2 | **-0.8** | **20.8** | 20.7 | 24.8 | 6.5 | 17.9 | 6.2 | 17.9 | 5.8 | 18.7 | 4.9 | 18.8 | 7.5 | 18.5 | 8.9 | 18.6 | -5.2 | 4.3 |
| Lakeside | 5.5 | 9.5 | 2.7 | 5.7 | -4.3 | 4.0 | **0.2** | **2.3** | 0.0 | 2.2 | -3.7 | 2.1 | -5.1 | 2.1 | 0.7 | 3.3 | -1.0 | 2.1 | -6.0 | 0.8 |
| Tiled | -2.4 | 9.2 | -4.9 | 7.8 | 6.0 | 12.4 | -3.8 | 8.4 | -2.8 | 8.5 | -5.1 | 8.6 | -5.9 | 8.7 | -4.7 | 8.6 | **-1.7** | **8.6** | -7.1 | 9.5 |
| K23 | 6.3 | 12.0 | -12.5 | 8.9 | -4.6 | 7.2 | 5.7 | 3.5 | 6.7 | 3.4 | 3.9 | 3.3 | **3.3** | **3.2** | 3.6 | 4.4 | 7.3 | 3.2 | 2.4 | 3.6 |
| K39 | 10.7 | 14.2 | **-0.5** | **12.2** | 12.9 | 13.6 | 4.0 | 11.2 | 3.8 | 11.2 | 4.2 | 11.4 | 3.6 | 11.3 | 5.3 | 11.0 | 7.5 | 11.3 | 6.1 | 9.2 |
| K56 | 0.3 | 5.9 | 1.5 | 11.3 | -1.3 | 8.9 | -0.3 | 1.3 | -0.5 | 1.0 | -1.1 | 1.8 | -1.9 | 1.7 | **-0.2** | **2.6** | 2.0 | 1.7 | 3.4 | 1.1 |
| K86 | 1.9 | 16.1 | -20.7 | 17.0 | -19.9 | 26.2 | 6.7 | 5.4 | 6.8 | 5.7 | 2.9 | 5.4 | **0.4** | **5.6** | 2.4 | 5.3 | 5.8 | 5.5 | -4.5 | 5.7 |
| K93 | -11.3 | 20.2 | -19.6 | 16.7 | -10.4 | 18.7 | -10.5 | 15.8 | -10.7 | 15.8 | -11.6 | 16.2 | -12.0 | 16.1 | -11.8 | 16.3 | **-8.3** | **16.1** | -9.4 | 17.9 |
| Shadows | 21.5 | 9.8 | -4.0 | 7.3 | **2.7** | **4.7** | 14.4 | 2.7 | 13.9 | 2.7 | 10.7 | 2.6 | 9.2 | 2.6 | 14.0 | 3.1 | 13.7 | 2.6 | 10.3 | 6.8 |

increase this threshold by 1 pixel each for the corresponding datasets and search again. If multiple networks are found within the threshold, we select the one with lowest overall position error. The median and standard deviation of position errors of the best networks for each colour model with the LCNN, MCNN and AlexNet architectures is presented in tables 4.4, 4.5 and 4.6 respectively. The median and standard deviation of width errors of the best networks for each colour model with the LCNN, MCNN and AlexNet architectures

is presented in tables 4.8, 4.9 and 4.10 respectively. Tables 4.7 and 4.11 respectively show the same for position and width errors of the bench-mark computer vision method (ASC) based on colour statistics. Cells with bold text highlight the lowest error across all colour spaces for each dataset.

Table 4.6: Median and Standard Deviation of position error (in pixels) of the modified AlexNet for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

| Dataset | RGB | | HSV | | Lab | | YUV | | YCbCr | | CbCra | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | med | std | med | std | med | std | med | std | med | std | med | std |
| R.Track | **-0.74** | **2.33** | 4.05 | 2.48 | 1.26 | 1.29 | 2.32 | 1.52 | 1.10 | 1.47 | 2.11 | 1.26 |
| Llan. | 2.61 | 5.35 | 9.71 | 5.40 | **-0.02** | **5.25** | 1.14 | 5.31 | 1.82 | 5.34 | 1.50 | 5.30 |
| W.Farm | -3.17 | 6.95 | -2.13 | 6.76 | -5.68 | 6.67 | **-1.85** | **6.63** | -2.09 | 6.67 | -3.74 | 6.78 |
| Rugged | -3.74 | 8.91 | 4.08 | 8.42 | -2.08 | 8.95 | -2.47 | 8.87 | -2.15 | 9.01 | **-0.53** | **9.02** |
| Rain | -4.17 | 10.64 | **-1.05** | **9.51** | -4.74 | 10.56 | -4.65 | 10.45 | -3.98 | 10.5 | -3.15 | 10.71 |
| Footpath | -3.10 | 8.36 | 3.57 | 8.34 | **-0.02** | **7.59** | 0.09 | 7.46 | -0.60 | 7.55 | -0.17 | 7.67 |
| MISC | -3.42 | 27.80 | -5.71 | 27.18 | -1.56 | 26.84 | **-0.41** | **26.9** | -2.13 | 27.02 | 0.80 | 27.23 |
| CMU | -14.72 | 13.59 | -14.97 | 14.43 | -11.20 | 12.07 | -12.03 | 12.65 | -12.07 | 12.65 | -9.31 | 12.40 |
| K87 | 6.87 | 7.56 | **0.34** | **7.58** | 1.92 | 7.43 | 2.10 | 7.27 | 3.70 | 7.07 | 0.99 | 7.40 |
| K59 | 11.66 | 2.67 | 1.99 | 3.21 | 1.09 | 1.03 | 1.37 | 1.50 | 3.53 | 1.28 | **0.58** | **1.17** |
| K34 | 12.99 | 18.36 | 6.48 | 19.22 | 6.04 | 18.61 | 6.38 | 18.48 | 8.20 | 18.35 | **5.70** | **18.42** |
| Steep | -5.85 | 5.69 | **1.74** | **6.10** | -4.67 | 5.55 | -5.08 | 5.42 | -4.45 | 5.67 | -3.72 | 5.61 |
| Lakeside | -2.57 | 1.87 | **0.24** | **1.97** | -4.30 | 1.75 | -2.61 | 1.79 | -2.06 | 1.75 | -3.23 | 1.79 |
| Tiled | -7.59 | 9.29 | **2.20** | **8.14** | -5.70 | 8.61 | -5.98 | 8.38 | -5.77 | 8.64 | -4.23 | 8.64 |
| K23 | 13.75 | 4.38 | **2.83** | **4.20** | 3.66 | 3.23 | 5.63 | 3.52 | 7.81 | 3.15 | 4.57 | 3.25 |
| K39 | 13.08 | 11.87 | 6.06 | 10.62 | 4.93 | 11.23 | 7.69 | 10.76 | 9.04 | 11.06 | **4.14** | **11.35** |
| K56 | 5.44 | 2.85 | 0.58 | 3.82 | **-0.62** | **1.78** | 1.28 | 1.52 | 2.24 | 1.82 | -0.88 | 1.36 |
| K86 | 11.65 | 7.18 | 6.10 | 5.41 | 3.75 | 5.72 | 3.99 | 6.27 | 5.02 | 6.14 | **3.46** | **5.69** |
| K93 | **-4.66** | **16.87** | -8.82 | 16.69 | -11.27 | 16.34 | -9.20 | 15.7 | -9.58 | 16.98 | -11.67 | 16.22 |
| Shadows | 14.02 | 2.77 | 14.25 | 3.07 | 10.12 | 2.67 | 11.33 | 2.60 | **1.10** | **1.47** | 11.48 | 2.66 |

From these tables, we can firstly observe that in terms of the position parameter, there is no combination of colour model and network architecture that can conclusively be said to be 'best suited' for all the datasets and by extension for application in outdoor trials. We find that no such combination provides a median error below 5 pixels for all 20 test environments (even though this is a somewhat arbitrary standard). The same can be said for the detection results of the ASC benchmark method, as no single colour model can be said to be globally applicable. A similar conclusion was reached at in [91], where the ASC method is presented and similarly evaluated offline in 6 of the 20 test datasets (Running Track, Llanbadarn, Shadows, Footpath, Wind Farm and Lakeside). However beyond this, a number of our trained networks indeed display very low errors in the majority of the datasets (close to 100 % accuracy in some cases). Moreover considering variations within each network

architecture we observe that overall detection accuracy does alter according to the colour model being used. This can be attributed to the manner in which colour information is represented in these models. For example, *RGB* does not distinguish between luminance and chrominance, while other models either have a dedicated channel for brightness or in the case of *HS*, *UV*, *ab*, *CbCr*, *CbCra* do not contain any luminance information (i.e. only chrominance). Indeed *YUV* and *YCbCr* which are the most similar among the colour models, show a much lower degree of disparity in performance. This variability across colour models is however somewhat less pronounced for AlexNet's position prediction (table 4.6) where results are more uniform (including *RGB*).

Table 4.7: Median and Standard Deviation of position error (in pixels) of ASC for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

| Dataset | RGB | | HSV | | HS | | lab | | ab | | YUV | | UV | | YCbCr | | CbCr | | CbCra | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std |
| R.Track | -4.0 | 22.4 | **1.0** | **1.1** | 2.0 | 1.0 | 0.0 | 19.5 | 3.0 | 1.6 | 2.0 | 1.2 | 2.0 | 0.9 | 2.0 | 1.2 | 2.0 | 0.9 | 2.0 | 4.1 |
| Llan. | 2.0 | 10.6 | **1.0** | **7.2** | 3.0 | 4.3 | 1.0 | 7.6 | 3.0 | 5.0 | 1.0 | 8.2 | 4.0 | 5.3 | 1.0 | 8.2 | 5.0 | 5.2 | 8.0 | 8.2 |
| W.Farm | 18.0 | 13.8 | 2.0 | 5.5 | 51.0 | 26.3 | **0.0** | **2.0** | -1.0 | 3.5 | 0.0 | 3.9 | -4.0 | 4.4 | 0.0 | 3.9 | -4.0 | 4.5 | -3.0 | 3.9 |
| Rugged | 1.0 | 8.3 | 1.0 | 8.9 | 0.0 | 9.7 | 1.0 | 8.5 | 0.0 | 10.2 | 3.0 | 9.2 | **0.0** | **9.4** | 3.0 | 9.2 | 1.0 | 9.5 | 1.0 | 14.6 |
| Rain | -2.0 | 18.6 | 3.0 | 15.2 | **2.0** | **15.0** | 6.0 | 17.5 | 4.0 | 15.3 | 6.0 | 13.1 | 5.0 | 12.7 | 6.0 | 13.2 | 4.0 | 12.7 | -3.0 | 14.2 |
| Footpath | 1.0 | 16.6 | 2.0 | 7.6 | 5.0 | 3.2 | 2.0 | 8.4 | 4.0 | 18.8 | **2.0** | **8.0** | 3.0 | 4.0 | **2.0** | **8.0** | 3.0 | 4.0 | 4.0 | 3.2 |
| K87 | -1.0 | 27.1 | -5.0 | 41.6 | 18.0 | 32.9 | **-2.0** | **22.2** | -2.0 | 51.1 | 12.0 | 34.4 | 43.0 | 54.0 | 6.0 | 35.7 | 43.0 | 54.0 | 16.0 | 35.2 |
| K59 | 9.0 | 17.5 | **0.5** | **44.5** | -44.0 | 7.6 | 10.0 | 19.9 | 5.0 | 47.5 | -5.5 | 20.1 | -13.5 | 48.5 | -5.5 | 20.1 | -13.5 | 47.4 | 13.0 | 32.3 |
| K34 | 3.0 | 11.6 | 8.0 | 12.5 | -4.0 | 9.7 | **2.0** | **6.6** | -7.0 | 5.7 | -11.0 | 5.8 | -13.0 | 4.5 | -11.0 | 5.8 | -13.0 | 4.5 | -43.0 | 9.5 |
| Steep | 0.0 | 11.9 | 0.0 | 8.6 | 1.0 | 7.9 | 1.0 | 9.8 | 1.0 | 7.4 | -1.0 | 7.9 | **0.0** | **6.8** | -1.0 | 7.8 | 0.0 | 6.8 | 2.0 | 10.1 |
| Lakeside | -2.0 | 8.8 | 0.0 | 1.4 | 1.0 | 0.9 | 0.0 | 1.3 | 1.0 | 0.8 | **0.0** | **1.2** | 1.0 | 1.2 | 0.0 | 1.2 | 1.0 | 1.2 | 3.0 | 0.9 |
| Tiled | 12.0 | 11.8 | 14.0 | 13.7 | 19.0 | 11.7 | 10.0 | 9.3 | 9.0 | 9.3 | **6.0** | **10.3** | 24.0 | 11.8 | 6.0 | 10.3 | 24.0 | 11.8 | 7.0 | 8.0 |
| K23 | 24.0 | 14.2 | 7.0 | 13.1 | -3.0 | 17.0 | 2.0 | 2.6 | **1.0** | **2.7** | 2.0 | 7.8 | -6.0 | 7.8 | 2.0 | 7.8 | -7.0 | 7.6 | -1.0 | 3.1 |
| K39 | -8.0 | 11.0 | -4.0 | 7.6 | -21.0 | 14.5 | -3.0 | 3.9 | **-2.0** | **3.1** | -4.0 | 4.6 | -3.0 | 14.9 | -4.0 | 4.6 | -3.0 | 14.8 | -2.0 | 3.9 |
| K56 | -29.0 | 7.2 | -34.0 | 5.8 | -46.0 | 11.6 | **-21.0** | **6.8** | -22.0 | 7.7 | -23.0 | 12.0 | -23.0 | 11.4 | -23.0 | 12.1 | -23.0 | 11.4 | -21.0 | 12.0 |
| K86 | -2.0 | 4.0 | **-1.0** | **3.3** | -2.0 | 12.0 | -1.0 | 5.2 | 52.0 | 51.4 | 23.0 | 22.6 | -16.5 | 64.9 | 8.0 | 20.2 | -16.5 | 64.9 | 10.5 | 32.7 |
| K93 | -34.0 | 28.6 | -5.0 | 18.2 | **-3.0** | **13.3** | -6.0 | 16.6 | -6.0 | 18.8 | -35.0 | 28.6 | -28.0 | 34.1 | -33.0 | 27.9 | -28.0 | 34.1 | -8.0 | 20.4 |
| Shadows | 13.0 | 3.9 | 14.0 | 2.5 | 14.0 | 2.2 | 13.0 | 1.9 | 14.0 | 2.0 | 13.0 | 1.7 | 12.0 | 2.0 | 13.0 | 1.7 | 12.0 | 2.0 | **7.0** | **3.8** |

For position errors across all colour models and datasets it can be said that the AlexNet architecture with its deeper structure provides a somewhat higher level of accuracy on the whole compared to LCNN and MCNN. It seems to specialise better to most datasets, compared to LCNN and MCNN, with more instances of median errors below 5 pixels. However choosing an appropriate colour model with the LCNN (such as *lab* or *CbCra*) and MCNN (such as *lab*, *UV*, *YCbCr*) can give position errors comparable to those achieved with AlexNet. With regards to the use of a progressively deeper architecture (with an extra convolution layer) from the LCNN to MCNN there does not seem to be an observable impact on detection generalisation; for some colour models (*lab*, *CbCr*) there seems to be an improvement in detection accuracy with the shift from LCNN to MCNN. However interest-

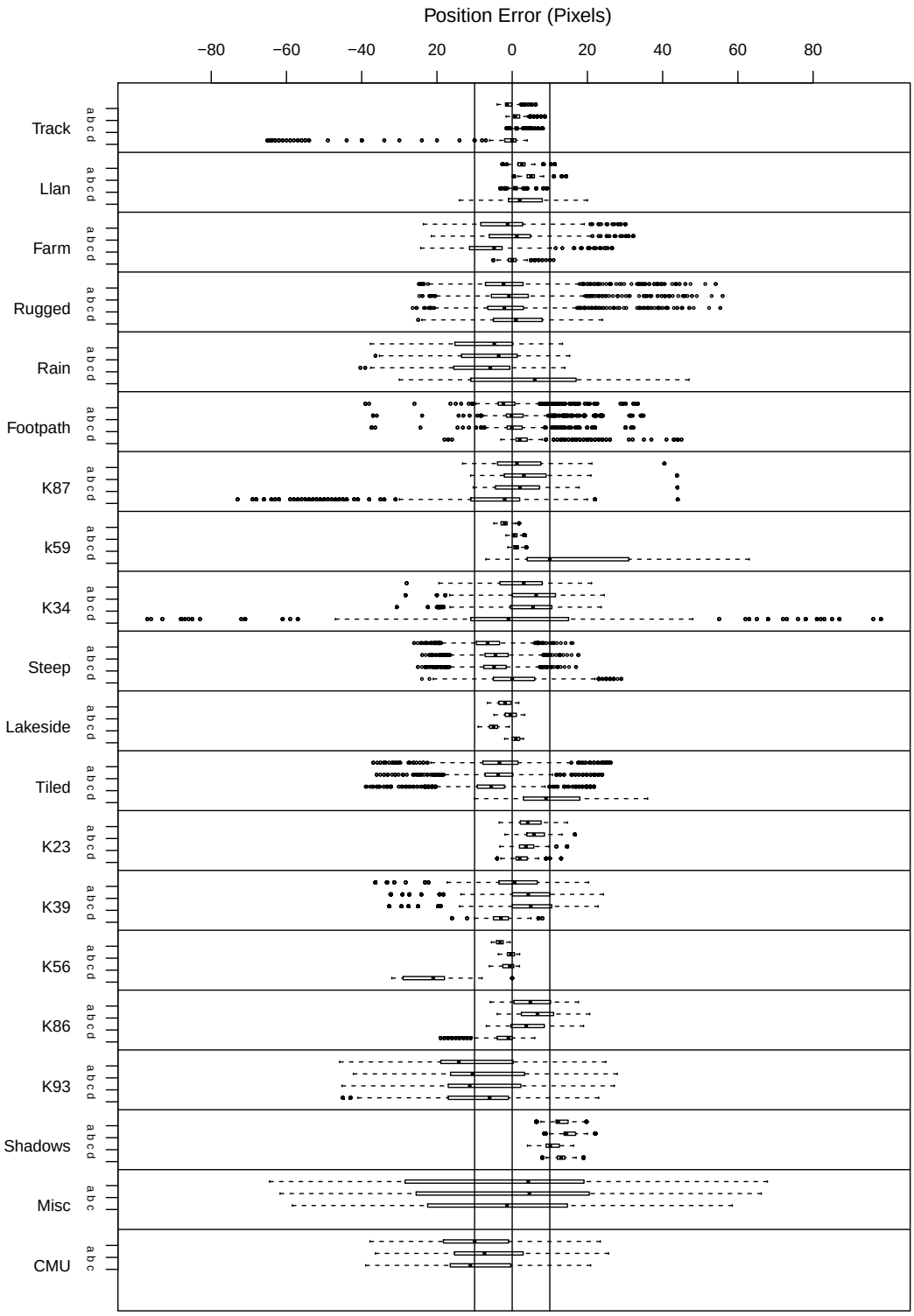



Figure 4-28: Boxplots comparing the position $(x)$ accuracy of the three convolutional network architectures (LCNN, MCNN and AlexNet) and the adaptive statistical colour-based (ASC) method for the lab colour model across all datasets. Plots for LCNN, MCNN, AlexNet and ASC for each dataset correspond to (a), (b), (c) and (d) respectively. Horizontal lines are drawn at the -10, 0 and 10 pixel error marks as visual aids.

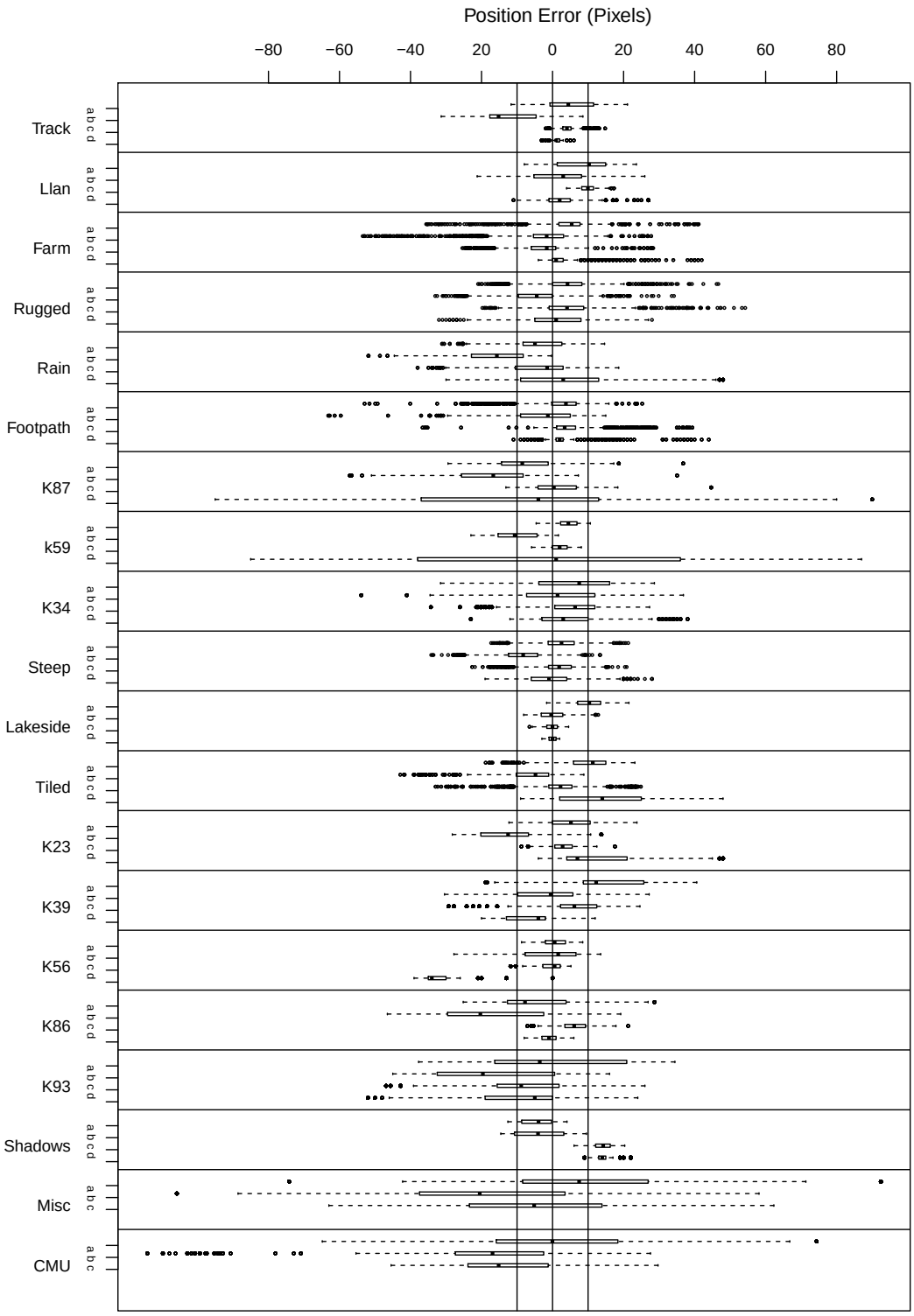



Figure 4-29: Boxplots comparing the position $(x)$ accuracy of the three convolutional network architectures (LCNN, MCNN and AlexNet) and the adaptive statistical colour-based (ASC) method for the HSV colour model across all datasets. Plots for LCNN, MCNN, AlexNet and ASC for each dataset correspond to (a), (b), (c) and (d) respectively. Horizontal lines are drawn at the -10, 0 and 10 pixel error marks as visual aids.

ingly with others there is either a deterioration (*HSV*, *HS*) or negligible differences in performance. It is interesting to observe that many of the trained networks achieve relatively low detection errors (albeit higher standard deviations) in the CMU and MISC datasets, where there is a great degree of environmental variability. As mentioned in Section 4.2.3 every image in MISC is representative of a different road environment. In general most trained networks performed poorly in the K93 dataset due to the inability to follow a turn to the right in the middle of the image sequence resulting in large negative errors for those frames (see Figure 4-30). Rather high detection offsets can also be observed for most cases in the dataset Shadows. This is due to the networks generally not including the half of the road covered in shadows in the detection, which while leading to errors compared with the ground truth would not cause the robot to go off-road.

The two boxplots in Figures A-10 and A-11, compare the position detection performance across datasets for two of the better performing colour models for the ASC method (*HSV* and *lab*) with corresponding networks of each architecture. The ASC method performed poorly in two KITTI datasets K56 and K59 (both of which contained urban roads) for both colour-models. The errors exhibited for both datasets are detrimental for navigation. In K59, the detected shape drifted off towards the right as the image-sequence progressed to include the pavement and not the road. In K56 the trapezoidal model was required to stay within the white lane demarcations (see Figure 4-31). Instead for both colour-models (more so for *lab*) the ASCs detected shape gradually shifted towards the adjoining lane on the left. As opposed to *HSV* (Figure A-11), the detection error distribution of the three network architectures with the lab colour model (Figure A-10) seem to be more consistent with one another.

It can also be observed that across all colour models (and both architectures), networks tend to predict the position parameter better than the width. While annotating the images to generate the teaching input for training, the width of the trapezoid has been arbitrarily set to accurately capture the road width. However, in case of very ill-defined roads, the definitions or road edges are fuzzy. For example, the overestimation of the width in many network/colour-model combinations in the dataset Steep is due to the inclusion of leaves on either side of the path within the road shape (see Figure 4-30). Similarly in the case of

Table 4.8: Median and Standard Deviation of width error (in pixels) of best LCNN for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

| Dataset | RGB | | HSV | | HS | | lab | | ab | | YUV | | UV | | YCbCr | | CbCr | | CbCra | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std |
| R.Track | 30.3 | 8.2 | **4.6** | **4.0** | 8.9 | 6.2 | 8.7 | 0.7 | 10.9 | 0.6 | 26.4 | 4.8 | 33.7 | 1.4 | 17.7 | 4.8 | 50.3 | 5.3 | 15.9 | 1.9 |
| Llan. | 49.2 | 14.0 | 23.7 | 10.9 | 9.6 | 11.0 | -3.3 | 10.8 | **-2.4** | **10.5** | 29.7 | 15.2 | 11.7 | 11.0 | 27.7 | 14.9 | 16.6 | 14.5 | -2.2 | 11.7 |
| W.Farm | -7 | 10.21 | **-1.0** | **7.2** | -2.3 | 7.7 | 13.0 | 3.1 | 21.7 | 2.5 | -3.8 | 8.8 | 16.2 | 8.4 | 0.9 | 10.0 | -1.6 | 6.5 | 2.6 | 6.1 |
| Rugged | -4.4 | 15.7 | 1.5 | 12.0 | 3.8 | 12.4 | -42.7 | 10.3 | -40.6 | 10.5 | **1.1** | **11.3** | -24.7 | 11.7 | -4.7 | 11.9 | -16.6 | 15.6 | -38.3 | 12.3 |
| Rain | -18.5 | 8.8 | -23.6 | 7.9 | -33.5 | 7.2 | -65.8 | 6.6 | -67.1 | 6.4 | **-15.5** | **8.8** | -45.7 | 8.2 | -22.1 | 9.3 | -21.9 | 11.9 | -58.7 | 9.7 |
| Footpath | 11.7 | 7.3 | 1.7 | 13.1 | -3.0 | 14.0 | -3.5 | 10.1 | **-1.8** | **9.4** | 4.3 | 6.8 | 13.1 | 12.0 | -5.4 | 9.1 | 20.4 | 21.0 | -3.2 | 11.6 |
| MISC | 9.8 | 32.1 | 3.1 | 18.5 | 3.0 | 20.6 | 9.1 | 16.4 | 13.1 | 16.0 | 16.6 | 26.2 | 14.8 | 19.8 | 15.0 | 22.6 | 8.6 | 22.5 | **2.6** | **18.6** |
| CMU | **-49.4** | **36.4** | -76.3 | 24.7 | -73.2 | 20.8 | -76.3 | 19.4 | -77.9 | 19.5 | -50.7 | 18.5 | -64.3 | 20.0 | -57.2 | 20.0 | -67.0 | 23.2 | -79.2 | 21.0 |
| K87 | 29.8 | 33.2 | 14.8 | 14.3 | 14.4 | 12.9 | 16.4 | 10.7 | 16.6 | 10.0 | 19.4 | 12.4 | 28.1 | 13.1 | **12.6** | **12.0** | 26.4 | 17.4 | 12.0 | 13.7 |
| K59 | **1.3** | **21.4** | 7.5 | 6.2 | 16.0 | 6.4 | 14.1 | 2.5 | 14.1 | 2.0 | 16.0 | 2.2 | 23.0 | 3.7 | 8.5 | 2.8 | 28.1 | 6.4 | 7.4 | 4.0 |
| K34 | 17.8 | 25.6 | 25.9 | 16.2 | 22.3 | 15.7 | 4.9 | 10.7 | 9.6 | 10.5 | 11.4 | 12.6 | 19.9 | 14.9 | 4.3 | 11.9 | 12.6 | 19.1 | **3.6** | **14.3** |
| Steep | 24.4 | 11.2 | 23.9 | 11.5 | 26.3 | 12.1 | -16.6 | 11.4 | -15.1 | 11.1 | 26.3 | 12.7 | **1.9** | **11.4** | 19.2 | 15.5 | 3.9 | 12.8 | -11.2 | 11.2 |
| Lakeside | -9.3 | 3.5 | 3.6 | 3.5 | 5.2 | 4.6 | 16.2 | 2.1 | 20.5 | 2.2 | -8 | 2.7 | 22.2 | 4.0 | 2.3 | 6.0 | **-2.4** | **3.5** | 7.6 | 4.0 |
| Tiled | -12.0 | 10.4 | **-9.8** | **12.1** | -18.1 | 10.8 | -60.3 | 8.1 | -62.7 | 7.9 | -14.3 | 11.0 | -42.8 | 7.9 | -19.6 | 12.4 | -24.2 | 9.3 | -51.7 | 8.4 |
| K23 | 6.5 | 14.1 | 8.9 | 8.6 | 11.1 | 6.1 | **-2.0** | **5.4** | 4.0 | 4.8 | 4.2 | 5.8 | 20.5 | 6.5 | -0.5 | 5.5 | 28.6 | 8.7 | 1.7 | 7.2 |
| K39 | -1.9 | 23.5 | 11.6 | 13.8 | 14.7 | 11.4 | 6.8 | 8.4 | 6.9 | 8.7 | 7.4 | 8.5 | 15.1 | 9.6 | 3.0 | 8.9 | 14.8 | 7.2 | **-0.7** | **9.2** |
| K56 | 28.0 | 22.1 | 26.3 | 11.0 | 27.0 | 9.6 | 9.6 | 2.8 | 12.1 | 2.3 | 15.7 | 3.3 | 24.8 | 3.3 | **8.4** | **2.9** | 29.2 | 7.2 | 9.5 | 3.1 |
| K86 | 24.2 | 20.3 | 20.3 | 15.3 | 21.2 | 25.5 | 20.7 | 5.4 | 22.2 | 4.8 | 26.5 | 7.8 | 32.6 | 6.2 | 18.8 | 7.6 | 27.8 | 13.3 | **16.7** | **7.1** |
| K93 | 11.2 | 22.8 | 10.4 | 12.4 | 14.3 | 13.5 | 2.7 | 10.9 | 7.1 | 10.2 | 13.1 | 13.5 | 19.5 | 10.6 | 5.8 | 11.9 | 22.6 | 15.0 | **4.0** | **10.8** |
| Shadows | 22.2 | 10.5 | 4.9 | 15.1 | -3.6 | 14.5 | -6.5 | 12.0 | **0.6** | **12.0** | -1.3 | 13.6 | 10.0 | 14.0 | -4.5 | 14.1 | -6.7 | 18.2 | -8.0 | 13.5 |

Lakeside and Wind Farm (which have very narrow roads), there is a tendency to include extra pixels on either side of the annotated ground-truth boundaries. It should be noted that networks have been selected purely on their ability to minimize position error and that similar degrees of accuracy across datasets and colour-models can also be observed with the width prediction of ASC. Refer to supplementary videos at `https://www.aber.ac.uk/en/cs/research/ir/dss/#road-driving` for a better understanding of the detection behaviour across different colour models and architectures. Appendix A gives boxplots with distribution of width errors for the *HSV* and *lab* colour models.

As mentioned previously initial experiments pointed to the fact that deep convolutional networks do not tend to generalise for images represented by colour models other than the one used for training images. This can be ascertained from the results presented in table 4.12, which shows median and standard deviation errors of MCNN networks predicting the position parameter with colour models different from the one they were trained in. The much poorer detection accuracy of these networks compared to when they are used with the colour model trained in, points to the fact that unlike the active vision controller from Chapter 3 it is advisable to train multiple networks corresponding to specific colour models.

Table 4.9: Median and Standard Deviation of width error (in pixels) of best MCNN for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

| Dataset | RGB | | HSV | | HS | | lab | | ab | | YUV | | UV | | YCbCr | | CbCr | | CbCra | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std |
| R.Track | 34.3 | 7.9 | 15.4 | 4.1 | 32.8 | 1.9 | 7.8 | 0.6 | 9.7 | 0.6 | **7.5** | **0.6** | 16.4 | 0.6 | 12.4 | 2.7 | 8.9 | 0.6 | x | y |
| Llan. | 50.4 | 16.3 | 8.9 | 11.5 | 14.7 | 10.7 | -4.7 | 10.5 | -3.5 | 10.6 | -6.2 | 10.4 | **3.3** | **10.5** | 4.0 | 10.5 | -4.7 | 10.4 | x | y |
| W.Farm | -8.0 | 13.1 | **-2.6** | **6.6** | -4.1 | 8.3 | 18.1 | 2.6 | 18.2 | 2.6 | 17.6 | 2.6 | 24.5 | 2.8 | 15.3 | 3.0 | 18.9 | 2.6 | x | y |
| Rugged | -5.9 | 15.4 | **2.5** | **12.0** | -4.6 | 11.6 | -43.7 | 10.5 | -42.2 | 10.4 | -44.0 | 10.4 | -34.2 | 10.6 | -39.7 | 10.7 | -42.3 | 10.4 | x | y |
| Rain | **-21.3** | **8.8** | -22.8 | 7.8 | -38.8 | 7.1 | -70.0 | 6.4 | -67.6 | 6.3 | -71.1 | 6.4 | -61.0 | 6.7 | -67.6 | 6.9 | -69.5 | 6.4 | x | y |
| Footpath | 10.6 | 10.8 | -3.2 | 12.7 | 11.3 | 20.7 | -4.8 | 9.5 | -2.8 | 9.8 | -5.2 | 9.5 | 3.6 | 9.6 | **-0.9** | **10.3** | -3.7 | 9.5 | x | y |
| MISC | 7.3 | 36.5 | **-1.5** | **20.9** | 14.8 | 22.8 | 10.2 | 15.9 | 10.4 | 16.1 | 9.2 | 15.7 | 16.8 | 15.8 | 7.3 | 15.6 | 10.8 | 15.8 | x | y |
| CMU | **-49.8** | **35.9** | -63.4 | 24.3 | -66.8 | 25.5 | -80.2 | 19.4 | -79.0 | 19.4 | -82.1 | 19.0 | -70.9 | 18.7 | -74.2 | 19.3 | -78.7 | 18.8 | x | y |
| K87 | 48.7 | 35.8 | 16.6 | 14.8 | 32.0 | 12.1 | 14.5 | 9.9 | 15.1 | 10.1 | **12.8** | **9.9** | 23.2 | 10.0 | 18.4 | 9.8 | 14.7 | 9.9 | x | y |
| K59 | -15.0 | 12.8 | 11.2 | 4.9 | 42.3 | 3.4 | 11.7 | 2.0 | 12.8 | 2.2 | 10.9 | 2.0 | 20.5 | 2.1 | 11.5 | 2.1 | 12.5 | 2.0 | **10.1** | **8.3** |
| K34 | 18.0 | 33.1 | 18.6 | 14.4 | 30.7 | 17.7 | **6.3** | **10.5** | 7.3 | 10.6 | 6.6 | 10.5 | 15.4 | 10.7 | 11.6 | 12.3 | 8.0 | 10.5 | x | y |
| Steep | 23.3 | 12.5 | 17.4 | 10.8 | 20.3 | 12.1 | -18.1 | 11.1 | -16.9 | 11.0 | -18.1 | 11.2 | **-8.0** | **11.3** | -13.3 | 11.6 | -16.4 | 11.2 | x | y |
| Lakeside | -9.0 | 2.7 | **4.9** | **2.7** | 12.2 | 9.5 | 17.8 | 2.2 | 17.9 | 2.1 | 16.6 | 2.2 | 24.1 | 2.2 | 12.7 | 2.3 | 17.9 | 2.2 | x | y |
| Tiled | -17.1 | 10.9 | -14.1 | 13.5 | -27.9 | 9.1 | -65.4 | 7.9 | **-6.3** | **7.8** | -66.1 | 7.9 | -55.9 | 7.9 | -61.9 | 8.3 | -64.5 | 7.9 | x | y |
| K23 | 18.2 | 21.5 | 11.2 | 7.0 | 27.3 | 4.9 | **0.3** | **4.8** | 1.9 | 4.9 | 0.0 | 4.7 | 9.9 | 4.7 | 5.5 | 5.6 | 1.5 | 4.6 | x | y |
| K39 | -6.8 | 25.8 | 11.7 | 8.0 | 32.2 | 9.0 | 4.6 | 8.7 | 5.3 | 8.5 | 4.1 | 8.7 | 13.5 | 8.6 | 4.6 | 8.3 | 5.5 | 8.7 | **3.9** | **7.1** |
| K56 | -16.0 | 20.3 | 19.9 | 5.7 | 36.3 | 7.1 | 9.2 | 2.5 | 10.3 | 2.3 | 9.1 | 2.6 | 18.2 | 2.5 | **7.0** | **2.2** | 10.5 | 2.6 | x | y |
| K86 | 36.2 | 32.2 | 24.8 | 13.1 | 31.7 | 22.2 | **18.3** | **4.8** | 20.8 | 4.8 | 18.4 | 5.0 | 27.8 | 5.1 | 21.7 | 5.2 | 20.1 | 5.0 | x | y |
| K93 | 12.2 | 24.6 | 14.1 | 10.2 | 32.9 | 11.7 | 3.3 | 10.3 | 4.4 | 10.2 | 3.9 | 10.1 | 13.0 | 10.1 | **1.5** | **10.4** | 5.5 | 10.2 | x | y |
| Shadows | 34.8 | 10.5 | 4.3 | 11.1 | **0.8** | **17.8** | -2.6 | 11.9 | -2.2 | 11.9 | -3.1 | 12.0 | 5.0 | 12.0 | 8.2 | 11.3 | -1.6 | 12.0 | x | y |

(a) (b) (c) (d)

Figure 4-30: Frames showing: (a) detection from CNN not including shadowed region in the road in Shadows, (b) failure to follow sharp turn in K93, (c) underestimation of width by not including leaves in road boundaries in Steep, (d) underestimation of width in Rain. The annotations in white correspond to the ground-truth and those in white to the CNNs detection output.

(a) (b)

Figure 4-31: Frames showing ASC detection failures in K59 (a) and K56 (b) datasets.

Table 4.10: Median and Standard Deviation of width error (in pixels) of the modified AlexNet for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

| Dataset | RGB | | HSV | | Lab | | YUV | | YCbCr | | CbCra | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | med | std | med | std | med | std | med | std | med | std | med | std |
| R.Track | **3.23** | **2.75** | 17.83 | 2.65 | 17.03 | 0.72 | 18.42 | 1.10 | 19.28 | 1.31 | 18.42 | 0.68 |
| Llan. | -7.18 | 11.01 | 5.41 | 11.45 | 3.31 | 10.57 | **1.42** | **10.79** | 2.17 | 10.83 | 4.57 | 10.48 |
| W.Farm | **10.19** | **3.63** | 12.83 | 3.99 | 25.46 | 2.67 | 15.74 | 2.90 | 16.42 | 3.01 | 25.41 | 2.76 |
| Rugged | -13.67 | 11.23 | **-6.65** | **12.02** | -33.72 | 10.53 | -25.67 | 10.58 | -25.39 | 10.68 | -33.30 | 10.46 |
| Rain | -49.78 | 8.40 | **-34.07** | **9.01** | -60.60 | 6.59 | -54.95 | 6.60 | -54.71 | 6.59 | -59.96 | 6.48 |
| Footpath | **-0.36** | **8.83** | 3.39 | 11.28 | 4.70 | 9.65 | 6.73 | 9.34 | 5.94 | 9.13 | 5.04 | 9.78 |
| MISC | 26.01 | 16.75 | 13.90 | 16.79 | 18.04 | 15.75 | 15.91 | 16.77 | 16.56 | 17.09 | 18.29 | 15.79 |
| CMU | -59.80 | 17.89 | -65.88 | 18.79 | -72.68 | 19.05 | -71.45 | 17.41 | -71.07 | 17.27 | -71.17 | 18.69 |
| K87 | **12.18** | **14.89** | 19.99 | 11.77 | 22.59 | 10.30 | 16.40 | 9.76 | 14.84 | 10.23 | 23.86 | 10.24 |
| K59 | **3.30** | **6.47** | 8.97 | 2.93 | 19.59 | 2.18 | 10.22 | 3.01 | 9.18 | 3.11 | 21.08 | 2.12 |
| K34 | **0.62** | **14.03** | 9.59 | 11.65 | 15.37 | 10.57 | 7.82 | 11.0 | 6.12 | 10.77 | 16.12 | 10.53 |
| Steep | 10.51 | 11.45 | 14.0 | 10.52 | -8.0 | 11.39 | -0.49 | 11.76 | **-0.22** | **11.59** | -7.87 | 11.11 |
| Lakeside | 4.94 | 3.72 | **5.06** | **2.80** | 25.56 | 2.18 | 16.64 | 2.57 | 18.08 | 2.63 | 26.23 | 2.20 |
| Tiled | -32.99 | 9.20 | **-23.15** | **13.03** | -55.58 | 8.12 | -48.11 | 9.10 | -47.93 | 8.61 | -55.20 | 7.86 |
| K23 | 7.66 | 8.81 | 6.20 | 5.21 | 10.0 | 4.89 | 1.39 | 4.85 | **-1.11** | **5.25** | 10.48 | 4.74 |
| K39 | -11.09 | 10.13 | 4.65 | 9.89 | 12.18 | 8.64 | 2.53 | 8.55 | **0.16** | **8.42** | 14.07 | 8.69 |
| K56 | **-1.85** | **6.12** | 16.38 | 5.07 | 17.98 | 2.08 | 10.24 | 3.51 | 8.71 | 2.80 | 19.03 | 2.43 |
| K86 | 20.61 | 11.18 | 24.87 | 7.04 | 28.18 | 5.35 | 20.35 | 6.55 | **19.56** | **7.30** | 28.21 | 4.93 |
| K93 | **-1.78** | **9.55** | 9.44 | 10.01 | 12.86 | 10.27 | 5.12 | 9.85 | 4.62 | 9.74 | 13.20 | 10.22 |
| Shadows | -12.01 | 10.74 | -0.64 | 11.61 | 5.75 | 12.28 | 2.77 | 12.77 | **0.27** | **12.52** | 5.90 | 12.11 |

Table 4.11: Median and Standard Deviation of width error (in pixels) of ASC for each colour model across all datasets. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

| Dataset | RGB | | HSV | | HS | | lab | | ab | | YUV | | UV | | YCbCr | | CbCr | | CbCra | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std | med | std |
| R.Track | -3.0 | 8.0 | -1.0 | 3.0 | 1.0 | 3.0 | 4.0 | 6.2 | 5.0 | 4.2 | -2.0 | 2.8 | **1.0** | **2.5** | -2.0 | 2.8 | 1.0 | 2.5 | 5.0 | 0.6 |
| Llan. | -8.0 | 14.2 | -2.0 | 11.7 | 11.0 | 8.9 | **-3.0** | **13.0** | 11.0 | 9.6 | -4.0 | 13.0 | 16.0 | 10.2 | -4.0 | 13.0 | 16.0 | 10.2 | 23.0 | 10.5 |
| W.Farm | 32.0 | 12.9 | 7.0 | 6.8 | 15.0 | 11.6 | **6.0** | **2.9** | 7.0 | 7.0 | 7.0 | 5.2 | 13.0 | 8.5 | 7.0 | 5.2 | 13.0 | 8.0 | 9.0 | 2.6 |
| Rugged | -52.0 | 13.5 | -49.0 | 13.5 | -44.0 | 13.6 | -50.0 | 13.5 | -43.0 | 14.7 | -49.0 | 12.2 | **-41.0** | **13.1** | -49.0 | 12.4 | -41.0 | 13.1 | -43.0 | 10.5 |
| Rain | -77.0 | 16.4 | -76.0 | 13.0 | **-72.0** | **14.1** | -77.0 | 12.7 | -75.0 | 12.6 | -78.0 | 11.1 | -74.0 | 12.1 | -78.0 | 11.1 | -73.0 | 11.9 | -78.0 | 6.4 |
| Footpath | -4.0 | 12.9 | -4.0 | 10.7 | 3.0 | 7.1 | -4.0 | 12.2 | **0.0** | **8.5** | -4.0 | 11.0 | -1.0 | 8.2 | -4.0 | 11.0 | -1.0 | 8.2 | -4.0 | 9.6 |
| K87 | **6.5** | **17.7** | 12.0 | 19.2 | 30.0 | 21.4 | 14.0 | 16.9 | 31.0 | 21.4 | 11.0 | 17.6 | 17.5 | 21.2 | 11.0 | 19.3 | 17.5 | 21.2 | 38.0 | 10.1 |
| K59 | **8.0** | **17.1** | 28.5 | 12.8 | 44.0 | 9.9 | 21.0 | 18.8 | 40.0 | 21.2 | 18.0 | 21.2 | 16.0 | 17.9 | 18.0 | 21.2 | 15.5 | 18.1 | 28.0 | 2.1 |
| K34 | 40.0 | 17.5 | 65.0 | 6.5 | 68.0 | 5.8 | **22.0** | **8.9** | 40.0 | 8.5 | 32.0 | 8.2 | 29.0 | 7.3 | 32.0 | 8.2 | 29.0 | 7.3 | 37.0 | 2.1 |
| Steep | -32.0 | 11.8 | -29.0 | 12.1 | -25.0 | 12.0 | -26.0 | 12.5 | **-17.0** | **13.6** | -29.0 | 11.8 | -25.0 | 12.1 | -29.0 | 11.6 | -25.0 | 12.1 | -21.0 | 11.2 |
| Lakeside | 3.0 | 10.9 | **0.0** | **2.4** | 2.0 | 2.7 | 0.5 | 2.4 | 2.0 | 2.5 | 1.0 | 2.1 | 2.0 | 3.6 | 1.0 | 2.1 | 2.0 | 3.6 | 3.0 | 2.2 |
| Tiled | -73.0 | 11.4 | -75.0 | 11.2 | -64.0 | 12.1 | -66.0 | 12.1 | -64.0 | 12.3 | -74.0 | 11.4 | -64.0 | 12.6 | -74.0 | 11.5 | -64.0 | 12.6 | **-51.0** | **7.9** |
| K23 | 27.0 | 18.4 | 8.0 | 13.5 | 19.0 | 14.1 | **-2.0** | **6.4** | 2.0 | 6.8 | 7.0 | 12.1 | 22.0 | 14.1 | 7.0 | 12.1 | 20.0 | 15.3 | -9.0 | 4.8 |
| K39 | 8.5 | 21.3 | 7.0 | 14.6 | 36.5 | 13.2 | -1.0 | 9.0 | **1.0** | **8.1** | 5.0 | 13.0 | 17.5 | 13.3 | 5.0 | 13.0 | 17.5 | 13.2 | 5.0 | 8.7 |
| K56 | 30.0 | 9.9 | 34.0 | 7.8 | 39.0 | 7.3 | 13.0 | 7.7 | 15.0 | 7.9 | 15.0 | 7.0 | 18.0 | 8.8 | 15.0 | 7.0 | 18.0 | 8.8 | **14.0** | **2.6** |
| K86 | 3.0 | 5.7 | 2.5 | 5.9 | 7.0 | 16.6 | **1.5** | **11.4** | 14.0 | 18.3 | 9.0 | 20.3 | 17.0 | 13.5 | 8.5 | 20.7 | 18.0 | 14.1 | 11.5 | 5.0 |
| K93 | 15.0 | 13.1 | 11.0 | 15.5 | 9.0 | 16.0 | -2.0 | 19.4 | 4.0 | 16.9 | 27.0 | 16.9 | 32.0 | 17.9 | 28.0 | 16.6 | 32.0 | 17.9 | **-1.0** | **10.2** |
| Shadows | -12.0 | 12.3 | -14.0 | 10.7 | -13.0 | 11.2 | -14.0 | 10.3 | -13.0 | 9.5 | -13.0 | 11.0 | -11.0 | 8.9 | -13.0 | 10.9 | -11.0 | 8.8 | **-2.0** | **12.1** |

## 4.4 Robot Trials

After evaluating the road detection performance of the convolutional neural networks trained with different architectures and colour models (tabulated in Section 4.3) two networks of the MCNN architecture corresponding to the *lab* and *UV* colour models were chosen for

Table 4.12: Median and Standard Deviation of position error (in pixels) of the MCNN for networks evaluated with colour models different from what they were trained in across all datasets. The first colour model in each column heading is the one used originally for training. Negative values indicate the predicted position of the road-shape being to the left of that in the ground truth.

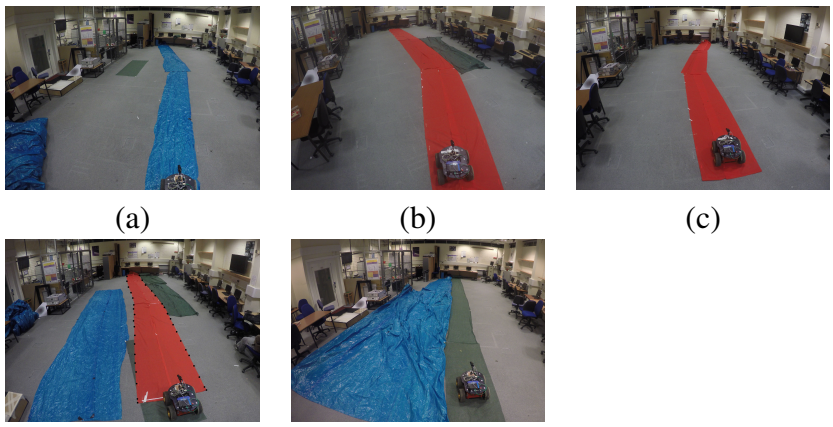
| Dataset | RGB-lab | | RGB-HSV | | HSV-RGB | | HSV-lab | | lab-RGB | | lab-HSV | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | med | std | med | std | med | std | med | std | med | std | med | std |
| R.Track | -15.5 | 1.7 | -19.8 | 4.6 | -31.1 | 3.9 | -10.0 | 2.6 | -16.4 | 4.5 | -16.0 | 4.4 |
| Llan. | -11.2 | 5.3 | -8.5 | 8.6 | -6.4 | 9.8 | 0.0 | 6.3 | 2.9 | 10.6 | 6.4 | 10.0 |
| W.Farm | -15.0 | 7.3 | -14.2 | 9.8 | -7.7 | 8.8 | -5.1 | 7.6 | -6.9 | 11.4 | -6.2 | 10.1 |
| Rugged | -17.0 | 9.1 | -27.1 | 10.3 | -28.6 | 9.4 | -13.9 | 9.2 | -21.3 | 9.8 | -13.5 | 9.5 |
| Rain | -19.4 | 10.5 | -26.4 | 10.7 | -26.4 | 10.8 | -15.5 | 10.2 | -18.6 | 11.5 | -14.6 | 11.7 |
| Footpath | -15.8 | 8.3 | -15.6 | 12.0 | -34.1 | 9.1 | -10.9 | 8.0 | -22.9 | 9.8 | -20.9 | 9.3 |
| MISC | -11.4 | 28.2 | -17.6 | 31.1 | -19.1 | 33.9 | -8.5 | 28.4 | -19.5 | 35.0 | -14.7 | 33.8 |
| CMU | -23.9 | 13.0 | -28.5 | 26.7 | -32.5 | 29.6 | -21.3 | 17.1 | -25.9 | 38.6 | -21.7 | 30.9 |
| K87 | -12.7 | 7.5 | -20.4 | 9.6 | -13.8 | 12.2 | -4.7 | 8.6 | -8.8 | 14.0 | -5.5 | 11.8 |
| K59 | -15.7 | 1.2 | -25.3 | 2.8 | -17.2 | 5.3 | -12.5 | 2.2 | -12.6 | 4.9 | -11.4 | 4.0 |
| K34 | -9.9 | 17.9 | -11.8 | 19.3 | -5.5 | 19.3 | 0.6 | 17.3 | 3.8 | 21.1 | 1.8 | 20.7 |
| Steep | -20.5 | 5.5 | -27.3 | 6.1 | -30.7 | 6.3 | -15.9 | 5.5 | -22.1 | 6.5 | -15.3 | 6.4 |
| Lakeside | -15.6 | 2.5 | -21.7 | 5.2 | -0.4 | 7.2 | -7.3 | 3.7 | -2.3 | 7.8 | 0.3 | 6.6 |
| Tiled | -19.7 | 8.7 | -22.0 | 9.4 | -23.4 | 9.1 | -13.7 | 8.4 | -15.6 | 9.5 | -9.5 | 9.9 |
| K23 | -9.7 | 3.7 | -11.9 | 8.2 | -14.8 | 8.2 | -0.9 | 5.4 | -6.1 | 10.1 | -8.7 | 8.2 |
| K39 | -12.6 | 11.4 | -13.5 | 9.3 | -8.2 | 14.0 | -5.9 | 11.0 | 4.8 | 13.0 | 4.2 | 12.4 |
| K56 | -17.0 | 1.2 | -25.6 | 3.0 | -11.3 | 7.4 | -8.5 | 1.8 | -10.1 | 5.2 | -10.4 | 5.5 |
| K86 | -10.0 | 5.9 | -17.3 | 11.2 | -13.0 | 12.3 | -3.1 | 7.0 | -10.9 | 12.7 | -8.5 | 11.2 |
| K93 | -26.6 | 16.2 | -32.2 | 20.4 | -25.1 | 16.0 | -21.5 | 17.3 | -28.2 | 17.4 | -26.0 | 16.7 |
| Shadows | -1.4 | 3.2 | -7.3 | 6.2 | 2.8 | 6.9 | 11.2 | 4.1 | 12.7 | 7.1 | 12.3 | 6.7 |

outdoor trials in 5 road environments. It is acknowledged that these were chosen arbitrarily and there were a number of suitable networks from either the LCNN or AlexNet architectures which could have potentially displayed higher or worse levels of performance. Results from off-line evaluation indicate that both the chosen networks (especially *lab*) could generalise to road environments different in colour distribution from those encountered in the training set. *UV* was selected to test the effect of using a network with two input channels (as opposed to three) and the effect of using a colour model with no luminance information. The MCNN architecture was chosen above LCNN with the hypothesis that the extra convolution could have resulted in the network learning more globalised, high-level features. While this could be extended to the AlexNet this was not chosen because of the extra computational associated with running this much larger network given available resources. Moreover there is the possibility of larger, more complex neural architectures

over-fitting/specializing to the types of environments, camera configurations and lighting levels encountered in the training set images. Although desirable, restrictions due to time and logistics associated with carrying out outdoor trials prevent the evaluation of further networks.

Before proceeding to the results of the outdoor road trials with the Pioneer 3-AT robot in Section 4.4.2, a set of preliminary trials with the same mobile platform was carried out with an LCNN network in 5 road segments created in an indoor laboratory. These tests confirmed the viability of road position predictions from a deep CNN trained using the methodology described in this Chapter, to control a robot in previously unseen environments. Success in these initial trials which were run with a 3 second input-output cycle ($\approx$), prompted the integration of a NVIDIA-TX1 GPU kit onto to the mobile platform to implement the trained convolutional networks in CUDA for real-time operation.

### 4.4.1 Preliminary indoor Trials



(a) (b) (c)

(d) (e)

Figure 4-32: Indoor test paths (a) Blue-Floor, (b) Red-Green, (c) Red-Floor, (d) Red-Blue, (e) Green-Blue. The path in image (d) has been manually annotated by black markers to show that the tarpaulin and netting on either side are not considered to be part of the road.

To ascertain the fact that road-detection outputs from convolution networks can be used to control an autonomous vehicle, we ported one of our networks with the LCNN architecture (operating in the *HSV* colour model) onto a Pioneer 3-AT robot. The robot's motion is controlled by a simple design rule which causes change in bearing based on the changes in

position values predicted by the network. It was observed that using a proportional control strategy, based on the assumption that a position output of 90 pixels corresponds to the center of the road, is not enough for guiding the robot in noisy real-world conditions. The networks tend to exhibit a systematic offset in the position prediction, which leads to the road being detected away from the center and closer to the edges. Therefore a simplistic derivative control loop which steers the robot, proportional to the difference in pixels of the current position prediction and that of the previous time step is employed. The steering angle $\theta$ is set by $\theta = \alpha(\zeta_0 - \zeta\_1)$, where $\zeta_0$ and $\zeta\_1$ are position detection outputs of the current and previous times-step respectively. The gain parameter $\alpha$ which would differ given the properties of the mobile platform being controlled, was set to a fixed value after a period of initial experimentation.

It should be noted due to hardware limitations (absence of a GPU) on the Pioneer 3-AT each update cycle took $\approx$ 3 seconds. It was for this reason we did not test the larger AlexNet which takes even longer to complete an update-cycle without GPU acceleration. We conducted a total of 25 trials across 5 different road environments that we created in an indoor laboratory. While not being representative of roads in the real-world these environments contain variations with respect to colour and geometry and they are significantly distinct from each other. More importantly, they were completely different to the type of roads that network had encountered as part of its training set. In addition the camera configuration used in these trials is also different to those used in training (see Table 4.2.2). All paths required the robot to make at least one turn to stay on-course. Since the purpose of these experiments is to test the network's ability to maintain track of unseen roads while on a highly noisy platform, we do not terminate a trial when the robot goes partly outside the boundaries. The trial is allowed to continue if the robot can correct its course and come back inside the road within 2 update cycles. If however the robot completely loses track of the road and travels off-course the trial is terminated.

âĂŃ The robot was successful in reaching the end of the road in 23 out the 25 trials (see table 4.13). As indicated by the divergence values in table 4.13, apart from environment Blue-Floor the robot's motion was at times quite oscillatory. It drifted towards either edge of the road at certain points and then had to make sharp turns to stay on course. It especially

Table 4.13: Summary of robot trial results across 5 environments. Divergence measures deviation from the center of the road.

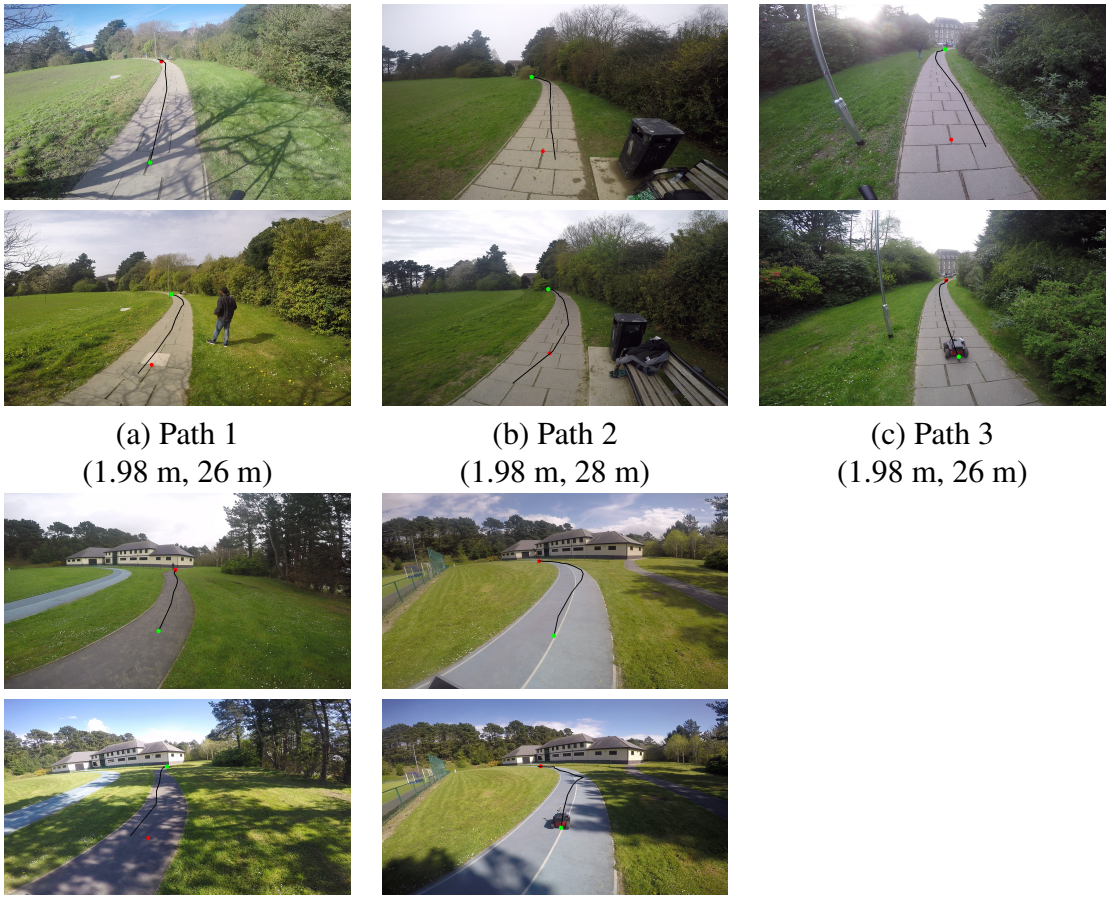
| Environment | Width (cm) | Divergence (cm) | | Success |
|---|---|---|---|---|
| | | mean | sd | num |
| Red-Floor | $101 \pm 15$ | 18.79 | 17.12 | 4/1 |
| Red-Green | $101 \pm 15$ | 23.10 | 18.34 | 5/0 |
| Blue-Floor | $78 \pm 18$ | 12.29 | 10.89 | 5/0 |
| Green-Blue | $83 \pm 7$ | 17 | 14.74 | 5/0 |
| Red-Blue | $101 \pm 15$ | 37.74 | 37.80 | 4/1 |

struggled to stay in the middle for environment Red-Blue where after navigating the first half of the road it kept turning towards the green netting (see Figure. 4-32) on the right. However the fact that the robot despite its rudimentary control system was able to execute turns and make constant adjustments is testament to the robustness of the network. Only a small minority of training images accounted for the sharp changes in detection the network had to repeatedly make to correct it's course during these trials.

### 4.4.2 Outdoor Trials

Implementing the networks in CUDA on a NVIDIA-TX1 mounted on top of the robot, resulted in execution time of $\approx 0.05$ seconds which is more than sufficient for real time control. Detection outputs from the TX1 board are sent using a TCP/IP protocol to the Pioneers on-board computer, which then executes the differential motion control loop. Similar to the preliminary indoor trials the steering angle $\theta = \alpha(\zeta_0 - \zeta_{-}1)$, where $\zeta_0$ and $\zeta_{-}1$ are position detection outputs of the current and previous times-step respectively.

Outdoor trials with the two selected MCNN networks (corresponding to *lab* and *UV*) were carried out in the same 5 test environments used for evaluating the active vision controller from the previous chapter (see Figure 4-33). For each environment, the robot undergoes a total of 20 trials, 10 for each colour model. For each set of 10 trials, at the sixth trial the robots starting position changes from the beginning to the end of the outdoor path, and consequently its direction of motion is inverted. A trial is successfully terminated when the robot traverses the entire length of the road without moving off the road boundaries. A trial is unsuccessfully terminated when either one set of wheels goes off the road bound-



(a) Path 1 (1.98 m, 26 m)

(b) Path 2 (1.98 m, 28 m)

(c) Path 3 (1.98 m, 26 m)

(d) Path 4 (1.81 m, 30 m)

(e) Path 5 (2.14 m , 28 m)

Figure 4-33: Outdoor environments. In each image, the black line refers to the robot's trajectory for a trial. The green circle shows the starting position, and the red circle denotes the end of the trajectory. Width (m) and length (m) of each path is indicated above each image. Images in the top and bottom rows correspond to trials carried out with the *lab* and *UV* colour models (respectively).

aries, or for exceeding the time limits set to 5 minutes. Since Path 1, Path 2 and Path 3 are sections of a longer public path, they all shared the same road surface but they differ in terms of non-road surfaces on either side (see Figure 4-33). It should be noted that there was also a degree of environmental variability even with trials in a particular path, with changing lighting, weather conditions and pedestrians serving as added complexities for the road-detection networks.

A summary of successful trials for both networks across the 5 test environments (as well as mean and standard deviation of the robots trajectory from the center of the road) is shown

Table 4.14: Number of successful trials per colour model for each outdoor environment (column 3); mean and standard deviation of the robot's divergence from the centre of the road (columns 4 and 5); mean and standard deviation of the time taken to complete a trial (columns 6 and 7).

| Env. | colour Model | Num. Succ. | Divergence (cm) | | Time (s) | |
|---|---|---|---|---|---|---|
| | | | mean | sd | mean | sd |
| Path 1 | lab | 9/10 | 9.5 | 58.7 | 114 | 25.3 |
| | UV | 9/10 | 24.9 | 72.0 | 112.8 | 37.6 |
| Path 2 | lab | 4/10 | 29.3 | 62.5 | 106 | 41.8 |
| | UV | 5/10 | 10.7 | 34.2 | 84.6 | 41.1 |
| Path 3 | lab | 7/10 | 13.7 | 53.3 | 95.4 | 16.6 |
| | UV | 6/10 | -3.2 | 32.9 | 114 | 47.5 |
| Path 4 | lab | 8/10 | -20.1 | 85.8 | 94.2 | 39.4 |
| | UV | 2/10 | -13.2 | 35.7 | 50 | 15.0 |
| Path 5 | lab | 8/10 | 94.2 | 138.8 | 136.6 | 69.5 |
| | UV | 6/10 | 88.9 | 95.4 | 119.14 | 34.1 |

in table 4.14. Although the network corresponding to the *lab* colour model was superior with 72 % successful trials over just 56 % for *UV*, neither could demonstrate the capability to generalise across all test environments. In path 2, where both networks performed poorly, the network trained with *UV* could successfully guide the robot 5 times in one direction, but consistently veered off-course towards a patch of grass in the reverse direction. The *lab* network failed 4 times in one direction and 3 in the other for this path, once again with the tendency to systematically drift towards the non-road surface around the half-way mark. Indeed this is a common theme with failures in Path 3 and Path 5 as well, with the robot going off the roads edges at roughly the same locations (towards the end of the course) in either direction. The two failures of the *lab* network in Path 4 were in the same direction, close to the end of the path. The especially poor performance of the other network ( *UV*) in this environment could be partly attributed to the weather conditions on the day the trials were being carried out which caused the appearance and reappearance of shadows on the road during the trials. Observation of detection streams during the trials seems to indicate that a majority of failures could be due to the network erroneously associating features in non-road surfaces with the road and/or not being sensitive to the features extracted in a sequence of frames which causes the robot to drift too far towards the edges. At times it was also observed that even if the network manages to regain tracking of the road, it is too

late to recover from the cumulative detection errors from previous time steps.

## 4.5 Conclusion

In this chapter we first present in Section 4.1 an experiment involving the development of very small convolutional neural networks through the evolutionary robotics approach (similar to the training scheme used in Chapter 3). Reviewing the performance of these networks showed a simpler feed-forward neural architecture performed comparably if not better in all test scenarios. Despite being a more complex neural architecture, when there are limitations on the number of layers and filters the performance benefits of such small CNNs are not apparent. Therefore when using the evolutionary robotics training approach, simpler feed-forward and recurrent neural models may be a more desirable option. CNNs become an advantageous choice when the architectures are 'deep' (more than 3 layers) with a sizeable number of filters in each layer.

Inspired by the impressive results attained by such 'deep' and 'wide' CNNs in various problem domains (see Section 2.3.2), we subsequently trained much deeper/larger networks through back-propagation by composing a dataset of annotated road images with the aim to encaspulate enough environmental variability that may lead to the emergence of a globalised feature representation. The road-following experiments presented in the previous section, show that such deep convolutional networks can be used to successfully navigate an autonomous vehicle/robot in noisy environments very different from what it was trained for. Compared with the benchmark colour based method (ASC), some of the CNNs are able to perform equally well if not better when evaluated off-line on road image datasets (see 4.3). In addition with the right colour representation, relatively shallower networks (LCNN, MCNN) can provide levels of detection accuracy at par with the deeper AlexNet architecture. However from the experimental results of 'off-line' and 'on-line' (robot trials) detection tests, no single CNN (corresponding to a combination of network architecture and colour model) could be shown to generalise across the spectrum of operational scenarios. Despite real-time processing of the update cycle, most instances of failure with these CNN models can be attributed to their inability to adapt or react to changes

along the course of the road. In most cases the reaction to changes in environment was too late to maintain the robot on the road, causing a ineffective course correction turns which eventually resulted in it going off-road. The reason for such failures are primarily due to an unbalanced training set, where the majority of images had the road positioned around the center of the image plane. Using data augmentation and standardisation techniques similar to those detailed in [13] to recreate more instances where the road position was displaced from the center, as well as image sequences with more dynamic lighting changes within them, could have resulted in CNNs that were more immune to such errors. It should also be noted at this juncture that even with GPU parallelisation, the ASC method is faster to execute (frequency of $\approx$30 Hz compared to $\approx$17 Hz for MCNN) and has also been shown to be able to autonomously navigate a robot on the outdoor test paths.

Contrary to other design based methods, convolutional networks whilst not easy to analyse and decompose into clear operational principles are somewhat less affected by direct human biases that sometimes limit the robustness and the capability of autonomous systems to operate in apriori unknown conditions. Having said this, it cannot be denied that the choices made behind the size and composition of the training dataset is one of the key factors behind the poor performance of the networks in certain outdoor test conditions. As shown by the experiment involving cross-dataset evaluation of deep CNNs published in [114], these models tend to specialise very well to the datasets used for training/validation but often perform poorly in other datasets despite the same labels and categories being present. Convolutional networks while exhibiting very low errors in 'closed dataset' evaluation may fail to transfer their performance to more complex real-world settings. Moreover as argued by the authors of [22] deep convolutional networks despite their size are still very simplistic compared to human/biological vision that is well adapted to the complexities of real-world operation. Indeed, from an applied, engineering perspective such as a robust, generalised road-detection system copying superficial features from the natural neural cortex may not be enough and 'might even be distracting from the goal of building a better vision system' in line with the standards observed in nature. Refer to the following concluding Chapter (Chapter 5) for continued discussion of this approach and the performance of the chosen networks in the outdoor trials in comparison with that the

active vision controller from Chapter 3.

# Chapter 5

# Conclusions and Future Work

This concluding chapter summarizes findings from the experiments involving the development of the dynamic active vision and deep convolutional neuro-controllers presented in Chapters 3 and 4 respectively. Limitations in methodology, training and implementation of both controllers, as well as proposed improvements for further performance improvements are discussed in Section 5.2. We provide a comparative evaluation of these two differing connectionist solutions to road-following in Section 5.3, discussing relative merits/demerits of the two approaches with regards to their performance in robot trials, practical/commercial viability of training/development and ease of integration into a larger self-driving control structure. We propose avenues for future experiments for incorporating additional capabilities (such as intersection detection, pedestrian avoidance etc.) into the types of neuro-controllers put forward in this thesis (Section 5.3.3). We also put forward a potential framework wherein deep-convolutional architectures can be integrated into an embodied dynamic 'active-vision' control system for autonomous driving. This can also be applied to other problem domains requiring similar levels of artificially intelligent sensorimotor control behaviour (Section 5.4).

## 5.1 Summary of Contributions

We outline below the overall findings and contributions arising from our attempts at developing two road-following solutions for robust generalised performance irrespective of

environmental conditions.

- Using the principles and methodologies from the field of evolutionary robotics, a continuous-time recurrent neural-network with a relatively small visual field (25 nodes) can successfully control a mobile platform along real and simulated roads whose visual composition is much different from those it was trained on. The network can transfer its sensorimotor control strategies on a road simulator to much more complex and noisy real-world scenes by dynamically altering the colour composition of its final visual input to extract meaningful regularities. This behavior emerges from unhindered exploration of a carefully designed evolutionary learning environment and additional selection pressures imposed through the fitness function which evaluates performance of 'individual networks' in a generation. Such a controller has been shown to navigate through varying lighting conditions and the presence of unseen features/objects in the environment, as well as performing sharp maneuvers to recover from errors without the need for any manual intervention.

- Convolutional neural networks whilst difficult to train for direct control of a mobile robot/vehicle can be trained 'off-line' on a dataset of annotated images to successfully predict the position and width of the road. When bench-marked on previously unseen sequences of road images with a computer vision algorithm that relies on differences in the Mahalonobis distance of colour channels, convolutional networks of different architectures have been shown to provide comparable if not superior levels of accuracy. Real-time predictions from a network trained in this manner can thus be used to control the motion of a mobile robot on a number of indoor and outdoor roads/paths.

- We can improve the performance of the 'evolved' dynamic active vision controllers through the use of colour channel combinations different to what it was trained under (i.e. *RGB*). We find that the highest levels of performance achieved is through a hybrid mixture of channels that are constituents of different colour models. Deep convolutional networks were on the other hand found to perform better when tested with images represented through the same colour space used for training. For both

approaches we observed there was no single colour combination that could be said to be suited for all operational scenarios. The authors of [91] arrived at a similar conclusion after evaluating their proposed road detection algorithm (refer to Chapter 2.1 for a review).

- The dynamic active vision approach seems to outperform the use of deep CNNs for road detection, in terms of more successful robot trials performed in the same set of trial roads (albeit at different time periods). Although the navigation strategy of the controller generated with the former approach cannot be described as 'linear' (i.e. the robot would make unexpected sharp turns, operate at different speeds in different segments of the road, etc.), the resultant sensorimotor behaviour was observed to be comparatively more robust and adaptive to dynamic changes on the road. See Section 5.3 for further comparison.

## 5.2 Limitations and Proposed Improvements

For implementation of both controller models the final developmental set-up was configured after a period of initial exploratory experiments with the aim of incorporating solutions to limitations of previous implementations reviewed in Chapter 2. There is however an almost exhaustive list of hyper-parameter configurations, if the entire spectrum of training algorithm and neural-architecture combinations pertaining to developing both controllers is taken into account. If we consider the dynamic active vision approach; visual field size, number of hidden network nodes, different colour mixing strategies, composition of learning/evaluation simulated environment, population size, evolutionary run time, etc. are some of the parameters and design choices that need to be in theory empirically evaluated. Similarly with the convolutional network experiments, alternate architectural and training hyperparameters, dataset composition, nature of image annotation, etc. could have been explored in further detail. Indeed there are a number of aspects that also need to investigated further with the approach of evolving relatively small convolutional controllers with direct motor control presented in Chapter 4.1. Thus despite our best attempts, further studies will need to be carried out for empirical evaluation of the effect of varying

some of these parameters and methodological configurations. The work presented in this thesis has been carried out with the aim of developing road-following solutions that can be evaluated in challenging real world conditions. With this is mind and given the limitations of allocated project completion time and available computational/hardware resources, we have explored various different solutions and tested different combinations of values for the large set of parameters of the system. The implementation details have been fixed from observing initialy exploratory experiments and reviewing related work in a manner we consider to be within the boundaries of reasonable scientific procedure. However, we cannot exclude that alternative methodological configurations would not result in the emergence of equally effective or even more robust road-following control.

### 5.2.1 Dynamic Active Vision Controller

Despite impressive performance in robot trials, there remains room for improvement with the dynamic active vision neuro-controller from Chapter 3. We identify below potential lines of research that can be explored in future experiments for a solution that can lead us closer to 100 % success across all conditions.

- **Image Pre-Processing**: The final input vector of the network is determined by dynamically mixing values from colour channels associated with the $\rho$,$\gamma$, and $\beta$ output parameters (see Chapter 2.1). The same mechanism can be used/extended to allow the network to dynamically choose pre-processing techniques to be applied on the raw camera input before it passes through the dimmensionality reduction stage to be reduced to 25 pixel-averaged grids for each colour channel. One possibility could be to have one or multiple additional output/feedback assigned to choose between pre-processing strategies such as Histogram of Oriented Gradients, Local Binary Patterns, Contrast Normalisation etc. (or indeed no pre-processing). Even without the aspect of network feedback and keeping the same network structure (i.e. no additional output nodes), it would be worthwhile investigating the effect of applying a pre-processing step on the raw pixel values of each colour channel.

- **Visual Receptive Field**: Each of the 25 input nodes of the neural-network controller

corresponds to a $100 \times 100$ grid of pixels on the raw image plane. However even with the same input vector length (i.e. 25 nodes) there are alternate strategies that can be explored for this resolution reduction step. Instead of overlaying grids over the entire image plane, a moving 'fovea' or grid of pixels could be used wherein the network can dynamically move, select and zoom in/out of a window of pixels over the image to extract desired visual detail on regions of interest. The network could thus in theory have additional output-feedback nodes which can control pan, tilt and zoom levels of the variable focus grid over the initial full-resolution input. 'Active' vision neuro-controllers such as the one presented in [93] for mobile navigation in an environment simulating the terrain of Mars, commonly employ this moving 'fovea approach'. It should be noted that such techniques involve additional network nodes which may lead to a larger solution search space for the evolutionary algorithm without necessarily resulting in performance improvements for real-world trials.

- **Multi-Stage Evolution**: For the experiments presented in this thesis, the neuro-controller is synthesised after a single 'continuous' evolutionary run in the 12 evaluation scenes detailed in Chapter 3. Although these scenes were designed to encourage the emergence of adaptive colour perception they are somewhat simplistic and not representative of the level of visual detail in outdoor conditions. A possible means to further 'bridge the reality gap' and improve transferability of the networks learning to real world scenes is to carry out a second stage of evolution using a population of controllers 'evolved' on the 12 virtual scenes mentioned above. The second 'evolutionary' run could involve longer evaluation trials over more complex visual scenes rendered using 'realistic' texture combinations.

### 5.2.2 Deep Convolutional Neural Networks

Our aim with developing the deep CNN was to produce a feed-forward passive vision system that could predict the position of the road irrespective of the environmental conditions, platform it was placed on etc. However despite impressive performance on 'off-line' datasets containing image sequences from various road conditions, outdoor robot trials

using selected networks' real-time predictions failed to generalise across all test road segments, displaying instances of repeatable systematic failure caused by particular features or aspects of the visual scene (see Chapter 4.5 for more detail).

- **Training Data**: The performance of the deep convolutional network trained for road-detection in Chapter 4 is strongly linked to the composition of its training dataset. Although we attempted to account for environmental variability by capturing/annotating images from multiple sources to constitute the training set, it could be argued that there was a degree of homogeneity to the data brought about by sequences of road images that were very similar to each other. Moreover only a small percentage of images in the overall set were representative of difficult scenarios such as sharp bends or the robot being positioned close the road edges which can be encountered during real-world trials. Further experiments with this approach would need to address this issue and attempt a different approach to constituting the training set from that used during the course of our experiments.

- **Colour Combinations**: As opposed to the former active vision approach, we found that deep CNNs trained for road-detection do not generalise to colour models different to those they were trained on. This resulted in us training multiple convolutional networks, each corresponding to a particular colour model. Observing experiments involving 'offline datasets' and 'online road trials' there was no single colour model that could be said to be 'best suited' for all environments. Results from the 'offline' evaluation however did indicate that most networks (corresponding to the colour models they were trained for) could specialise very well with low detection errors to one or more datasets. Even during outdoor robot control trials with the two MCNN networks receiving images encoded with the *lab* and *UV* colour models respectively; there were instances of either network repeatedly guiding the robot with success in operational scenarios where the other failed completely. With this in mind it is may be beneficial to explore hybrid colour combinations (besides *CbCra*), composed of individual colour channels from different models. Indeed with the dynamic active vision neurocontroller, the best performing colour combination (USH) was composed

of channels from the *YUV* and *HSV* colour models. Moreover there is also a case for testing convolutional networks with more than 3 input colour channels. For example a CNN could be trained with 6 input channels, thereby incorporating channels from two or more colour models into a single network. While this may result in a increase in number of training parameters, it would be relatively negligible and could potentially result in road-detection performance otherwise not possible with the use of fewer input colour channels.

- **Platform Specific Network and Unsupervised Annotation**: One of the main aims with the experiments involving deep convolutional networks was to develop a general-purpose road-detection system that can be ported onto any mobile-platform irrespective of its structural, kinematic properties and the positioning, configuration of the input camera. To achieve this images captured from a number of different sources were incorporated into the training data-set. An alternative to this approach could be to forgo the idea of having a global, general-purpose road-detection network and instead focus on training a network that always receives images from a specific camera configuration. In other words performance improvements can be expected if a network is designated to operate in a specific mobile platform; with the training dataset composed of images that correspond to the target robot or vehicle's camera properties and positioning. One possible approach to achieve this would be to collect training snapshots from driving around different roads/paths. Alternatively taking inspiration from the training scheme employed in [42], instead of manual annotation, training labels could be automatically generated from a computer vision method (such as the ASC). It would be interesting to see if a network that learns (partly or entirely) from somewhat noisy labels (generated by the ASC) can develop sensitivity to regularities that enable it to control the robot in conditions where the ASC detection method fails.

## 5.3 Comparative Evaluation

The two road following solutions proposed in thesis are somewhat difficult to compare given their methodological differences, with each one attempting a somewhat different ap-

proach to solving a visuo-cognition problem through connectionist models. Although it would be incorrect to label either approach as being better than the other (in the context of robust and generalised road following behaviour), the subsequent discussion in this section aims to provide the reader with a better understanding of the relative merits/demerits of the two.

### 5.3.1 Robot Control Performance

The only reference point for direct comparison of the two approaches presentend in this work is in analysing their respective performance in controlling the same mobile robot (Pioneer 3-AT) over a common set of 5 outdoor road segments (albeit over different times and weather/lighting conditions). The dynamic active vision neuro-controller from Chapter 3 was more successful in this regard, with 84% success in trials for the best performing colour combination (USH) compared to 72% for the MCNN network using the lab colour model. More importantly with the active vision controller we were able to demonstrate that with one of our chosen colour combinations (bUV and aSH being the other two), the controller could guide the robot with atleast 80% success across all test environments. This was not the case with the deep CNN where even the better of the two networks displayed rather poor detection behaviour in two road segments, with repeatable, systematic failures. However this is not conclusive proof of the superiority of the former approach as it is conceivable another network (corresponding to a different architecture, colour model combination) could have performed better than the two that were evaluated in the outdoor trials.

On the whole it was observed that the trajectory of the robot in trials with the deep CNN road detector was relatively more uniform and less 'sinuous' compared to when it was directly controlled by the active vision neurocontroller. This can be attributed to the separation of visual perception and motor control modules with the convolutional network based solution; with a 'hand-designed' control strategy dictating the wheel speeds based on road position predictions from the network. In contrast to this the neurocontroller from Chapter 3 integrates vision and action into a single control loop. This makes the motion

and trajectories of the robot when controlled by it harder to decompose and predict as it is closely linked to the levels of differential contrast perceived by the network. There were many instances during the trials with the active vision controller when it was observed that the robot would exhibit sudden changes in speed and spend many iterations of the input-output cycle where it would be almost stationary; trying to extract the desired regularity to proceed along the course. However this integrated approach to action and cognition was also responsible for the better adaptability of the system to different environments and complex dynamic changes in the environment. One of the key advantages with the active vision approach was that the robot could recover from wrong turns and navigation errors by making a sequence of sharp course corrections, a behavior that emerged from the evolutionary training phase in simulated road environments.

### 5.3.2 Computational Resources

For both solutions, there was a requirement for parallel computation to run processes that would otherwise result in unreasonable delays given a regular machine with limited cores. The active vision controller especially required the use of an off-site computing cluster (HPC-Wales) to execute multiple seeded evolutionary runs. However beyond the training phase the actual controller could be implemented in serial on the Pioneer 3-ATs embedded motherboard, without the requirement for special hardware. On the other hand it is possible to train on deep convolutional networks on a single server given the presence of a GPU module. However there was also a need to purchase and mount a GPU enabled processor (the NVIDIA-TX1) onto the Pioneer to process the deep CNNs input-output cycle fast enough for real-time robot control. While neither approach could be said to be independent of the need to incorporate specialised computing hardware, the active vision controller (while requiring parallel processing during its training phase) has the advantage of being implemented on a simplistic platform during real-world operation.

### 5.3.3 Integration into a Larger Control Structure

Road-Following/Detection is one among other tasks such as obstacle avoidance, path-planning, traffic-sign detection etc., a fully autonomous road-driving control-stack must account for. Therefore it is important to consider the ease and viability of integrating the two proposed road-following solutions into a larger, hierarchical control structure; wherein the road-following behaviour must aid and not interfere with other functionalities and overall system objectives. This is somewhat straightforward with the deep CNN solution, as this is essentially a reactive system which processes an image and predicts the roads position in the image frame. This can be seen as an independent module, easily integrated into a global control loop. However this is more challenging with the active-vision controller due to the motion control and visual perception being part of a single closed loop and the network directly controlling the robot's wheel speeds. While a framework could be conceived of where the network is activated and reset by the larger control loop when required, this can be somewhat impractical and lead to additional complications. Ideally it would be desirable to have a single closed-loop neurocontroller generated through the embodied evolutionary process which exhibits all the necessary behaviours required for navigating an autonomous vehicle/robot to the desired location.

However a set of experiments (beyond the remit of this thesis) conducted by us to incorporate the ability to deal with intersections into the active vision controller indicate that generating a network controller that can exhibit the range of behaviors required for fully autonomous driving is not viable with the methodological approach presented in Chapter 3. We incorporated an additional input node to the CTRNN network, the value of which is set based on pre-existing knowledge of the course. This value dictates whether the robot should take the left or right branch of the the next/up-coming intersection in the road, assuming all intersections only contained two branches. The virtual road environment was slightly modified for these experiments. While the same texture combinations were used to render the 12 road scenes, the shape of the course was changed to incorporate an intersection after an initial curved segment. In brief, results from differently seeded evolutionary runs showed that networks could learn to choose the correct intersection branch based on

the value of the associated input node in some scenes. However there was no single network across multiple evolutionary runs that could exhibit this across more than 7 of the 12 simulation scenes. In the remaining scenes, networks tend to go off-course at the beginning of the road segment which suggests that the evolutionary search algorithm could not converge at a global solution.

A practical and more viable alternative for developing a single neuro-controller in charge of multiple road-following related functionalities would be to consider the principle of behavioral decomposition. An example of which can be seen in hierarchical control framework presented in [31]; wherein a complex task is broken down into smaller-sub-tasks and an overall high-level controller is evolved to switch and activate controllers further down in the hierarchy. The lower level sub-tasks are either managed by controllers that have been independently evolved or 'hand-designed'. It was shown that the tasks that were solved using this approach of a hybrid control structure with a mixture of evolved and hand-designed components could be solved with the use of a single overarching neurocontroller. An example of how this idea can be applied to autonomous driving is as follows; a robot can be controlled by an evolved neurocontroller (akin to the one proposed in this thesis) as it travels along a road and then while approaching intersections autonomously switch to another control module that has either been programmed or evolved specifically for this sub-task.

## 5.4 Future Directions

In this study we have introduced two possible avenues of research that may warrant further exploration to develop a fully or near-fully autonomous road following solution that can be globally applied across all environmental scenarios. Indeed the exploration of each of the two solutions to road-following presented in this thesis can be significantly expanded independently as part of future studies incorporating new methodologies, taking into consideration the proposed improvements suggested in Section 5.2. This section details what we feel are some of the most promising future research approaches based on the cumulative findings of the experiments and theoretical principles presented in this work. In our

opinion the ability to navigate through complex environmental variability exhibited by the two controllers during our experiments may be significantly improved upon by a methodology that integrates the two somewhat distinct lines of study of evolutionary robotics and deep-learning. What if 'intelligent agents' were to be evolved with 'active perception' capabilities receiving high-level features detected by deep convolutional networks as their input stream? This is opposed to the norm of evolutionary active vision works employing relatively simplistic sensori apparatus, a $5 \times 5$ grid of averaged pixels in the case of the controller presented in Chapter 3.

One possible approach to implement such a controller would be to train CNNs on offline datasets, with a methodology similar to that described in Chapter 3. An active vision controller could then be evolved in a virtual environment under the tenets of evolutionary robotics of to dynamically control intensity levels of a CNNs input colour channels. Alternatively it could also be trained to from the final input vector by controlling the mixture of feature contributions from multiple CNNs (if computational resources are not a limitation). Besides it would also be worthwhile to investigate the development of a more complex control framework capable of multiple tasks beyond just road following, with the principles of behavioural decomposition discussed above in Section 5.3.3. A hybrid framework consisting of deep convolutional networks, hand-designed algorithms with the active control properties could be developed to seamlessly transition and integrate different behavioral functionalities to achieve high-level navigation goals (such as traveling between two points on a map) in a fully autonomous manner. On the whole, despite recent advances, biological vision (especially human or primate vision) is still many times more complex than artificial systems for tasks such as fully autonomous driving which requires robust perceptual discrimination in complex real world conditions. Instead of building systems based on simplified abstractions of specific traits found in brains of biological organisms; it might be worthwhile working towards a holistic integration of concepts like spatio-temporal hierarchy, modularity, active perception, embodied development, etc. all of which play a key factor in the development of the complex visual cognition behaviour observed in nature.

# Appendix A

# Supplementary Videos and Figures

**Links to supplementary videos of the robot being controlled in real and virtual environments with the active vision controller from Chapter 3.**

Figure A-1: To play the video, click on the image or use the following URL `https://www.youtube.com/embed/EtgPU-mwn94`. This video shows the 'evolved' neural network from Chapter 3 controlling the Pioneer 3-AT robot in 'Path 1' using the 'USH' colour model.

Figure A-2: To play the video, click on the image or use the following URL `https://www.youtube.com/embed/6XBtsxax5xk`. This video shows the 'evolved' neural network from Chapter 3 controlling the Pioneer 3-AT robot in 'Path 2' using the 'USH' colour model.

Figure A-3: To play the video, click on the image or use the following URL `https://www.youtube.com/embed/MKvPLvQHcbM`. This video shows the 'evolved' neural network from Chapter 3 controlling the Pioneer 3-AT robot in 'Path 3' using the 'USH' colour model.

Figure A-4: To play the video, click on the image or use the following URL `https://www.youtube.com/embed/hyr5J47V_w0`. This video shows the 'evolved' neural network from Chapter 3 controlling the Pioneer 3-AT robot in 'Path 4' using the 'USH' colour model.

Figure A-5: To play the video, click on the image or use the following URL `https://www.youtube.com/embed/Mhki09BN0YM`. This video shows the 'evolved' neural network from Chapter 3 controlling the Pioneer 3-AT robot in 'Path 5' using the 'USH' colour model.

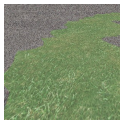


Figure A-6: To play the video, click on the image or use the following URL `https://www.youtube.com/embed/wn6YvTwEWCQ`. This videos shows an 'evolved' neural network from Chapter 3 controller navigating different virtual simulated road environments.

**Links to supplementary videos of 'offline' detection and online robot control trials with deep CNNs from Chapter 4.**

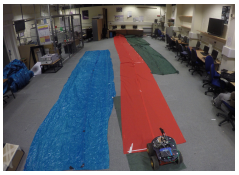


Figure A-7: To play the video, click on the image or use the following URL `https://www.youtube.com/watch?v=umx6Fr9Qe5o`. This video shows a deep CNN trained with the HSV colour model (from Chapter 4) controlling a robot in indoor paths.

Figure A-8: To play the video, click on the image or use the following URL `https://www.youtube.com/watch?v=qO3Iop-SwrY`. This video shows a trained deep CNN (from Chapter 4) detecting roads on 'off-line' datasets.

Figure A-9: To play the video, click on the image or use the following URL `https://www.youtube.com/watch?v=ueCxt-ZHII0`. This video shows a deep CNN trained with lab(from Chapter 4) controlling a robot in the five outdoor test paths.

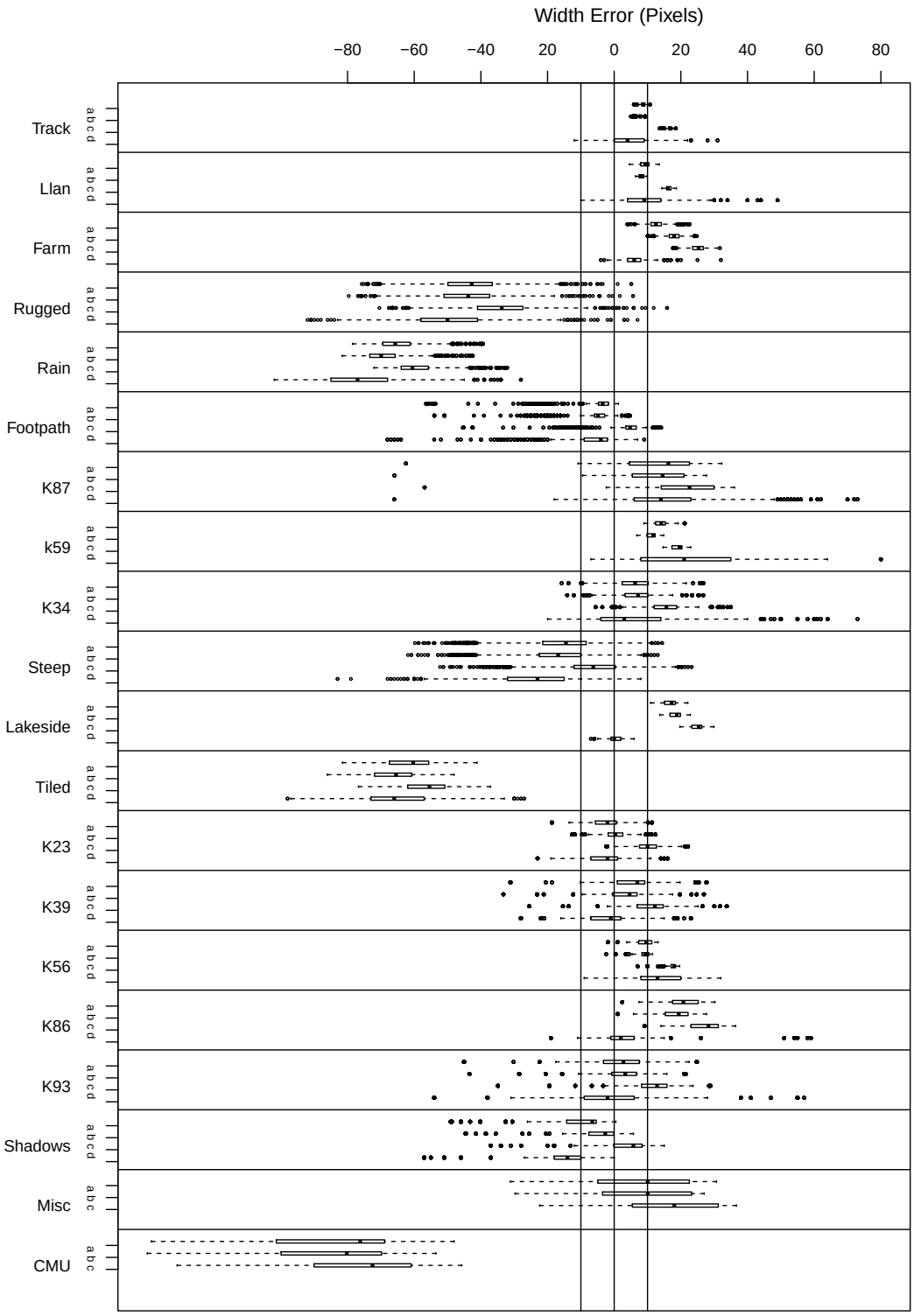



Figure A-10: Boxplots comparing the width $(x)$ accuracy of the three convolutional network architectures (LCNN, MCNN and AlexNet) and the adaptive statistical colour-based (ASC) method for the lab colour model across all datasets. Plots for LCNN, MCNN, AlexNet and ASC for each dataset correspond to (a), (b), (c) and (d) respectively. Horizontal lines are drawn at the -10, 0 and 10 pixel error marks as visual aids.

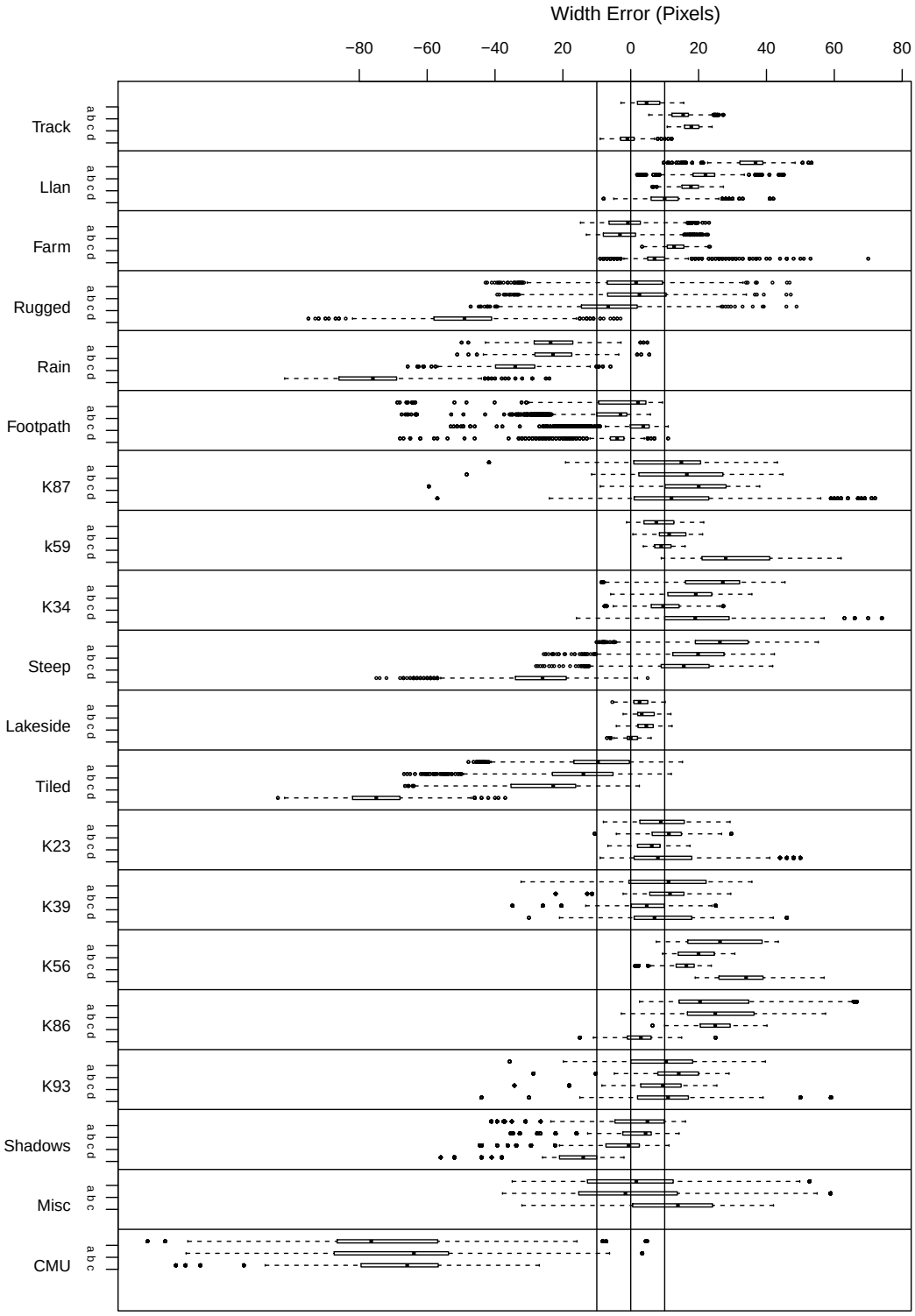



Figure A-11: Boxplots comparing the width $(x)$ accuracy of the three convolutional network architectures (LCNN, MCNN and AlexNet) and the adaptive statistical colour-based (ASC) method for the HSV colour model across all datasets. Plots for LCNN, MCNN, AlexNet and ASC for each dataset correspond to (a), (b), (c) and (d) respectively. Horizontal lines are drawn at the -10, 0 and 10 pixel error marks as visual aids.

# Appendix B

# Colour Model conversions from RGB

**HSV (Hue Saturation Value)**

Linear Transformation from RGB as follows

$M = max(R,G,B)$

$m = min(R,G,B)$

$C = M - m$

Keeping the above values in mind, the transformation to HSV from RGB is

$V = M$

$$H = 60^o \times \left\{ \begin{array}{c} undefined\ (if\ C = 0) \\ ((G-B) \div C)\ mod\ 6\ (if\ M = R) \\ ((B-R) \div C)\ +2\ (if\ M = G) \\ ((R-G) \div C)\ +4\ (if\ M = B) \end{array} \right\}$$

$$S = \left\{ \begin{array}{c} 0\ (if\ C = 0) \\ (C \div V)\ (otherwise) \end{array} \right\}$$

**YUV**

Linear Transformation from RGB as follows

$Y = 0.299R + 0.587G + 0.114B$

$U = 0.492(B - Y)$

$V = 0.877(R - Y)$

R,G,B values are normalised between 0 and 1

**YCbCr**

Linear Transformation from RGB as follows

$Y = 0.299R + 0.587G + 0.114B$

$Cb = 0.5 - 0.169R - 0.331G + 0.500B$

$Cr = 0.5 + 0.500R - 0.419G - 0.081B$

R,G,B values are normalised between 0 and 1

**lab (CIE $L^*a^*b^*$)**

Non-Linear Transformation from RGB as follows

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2.7690 & 1.7518 & 1.1300 \\ 1.0000 & 4.5907 & 0.0601 \\ 0.0000 & 0.0565 & 5.5943 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

From the the values of X,Y,Z the transformation to lab is

$L^* = 116 \times (Y \div Y_n)^{1/3} - 16$

$a^* = 500 \left[ (X \div X_n)^{1/3} - (Y \div Y_n)^{1/3} \right]$

$b^* = 200 \left[ (Y \div Y_n)^{1/3} - (Z \div Z_n)^{1/3} \right]$

**CbCra**

This is a hybrid colour model dervied from YCbCr and lab. There are no luminance channels in this model, instead it is comprised of the Cb, Cr channels from YCbCr and the a channel from lab.

# Bibliography


[1] Aloimonos, J.: Purposive and qualitative active vision. In: [1990] Proceedings. 10th International Conference on Pattern Recognition, vol. i, pp. 346–360 vol.1 (1990). DOI 10.1109/ICPR.1990.118128

[2] Aloimonos, J., Weiss, I., Bandyopadhyay, A.: Active vision. International Journal of Computer Vision **1**(4), 333–356 (1988). DOI 10.1007/BF00133571. URL `http://dx.doi.org/10.1007/BF00133571`

[3] Alon, Y., Ferencz, A., Shashua, A.: Off-road path following using region classification and geometric projection constraints. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 1, pp. 689–696 (2006). DOI 10.1109/CVPR.2006.213

[4] Alvarez, J., Gevers, T., LeCun, Y., Lopez, A.: Road scene segmentation from a single image. In: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds.) Proc. of the European Conf. on Computer Vision, pp. 376–389. Springer Berlin Heidelberg (2012)

[5] Álvarez, J., López, A.: Road detection based on illuminant invariance. IEEE Transactions on Intelligent Transportation Systems **12**(1), 184–193 (2011)

[6] Ampatzis, C., Tuci, E., Trianni, V., Dorigo, M.: Evolution of signaling in a multi-robot system: Categorization and communication. Adaptive Behavior **16**(1), 5–26 (2008)

[7] Apostoloff, N., Zelinsky, A.: Robust vision based lane tracking using multiple cues and particle filtering. In: Intelligent Vehicles Symposium, 2003. Proceedings. IEEE, pp. 558–563. IEEE (2003)

[8] Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Transactions on signal processing **50**(2), 174–188 (2002)

[9] Audigier, R., de Alencar Lotufo, R.: Relationships between some watershed definitions and their tie-zone transforms. Image and Vision Computing **28**(10), 1472–1482 (2010)

[10] Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)

[11] Barnes, D., Maddern, W.P., Posner, I.: Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy. CoRR **abs/1610.01238** (2016). URL `http://arxiv.org/abs/1610.01238`

[12] Beer, R.D., Gallagher, J.C.: Evolving dynamic neural networks for adaptive behavior. Adaptive Behavior **1**(1), 91–122 (1992)

[13] Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., Muller, U.: Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911 (2017)

[14] Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)

[15] Cadieu, C.F., Hong, H., Yamins, D.L., Pinto, N., Ardila, D., Solomon, E.A., Majaj, N.J., DiCarlo, J.J.: Deep neural networks rival the representation of primate it cortex for core visual object recognition. PLoS computational biology **10**(12), e1003,963 (2014)

[16] Chapuis, R., Aufrere, R., Chausse, F.: Accurate road following and reconstruction by computer vision. IEEE Transactions on Intelligent Transportation Systems **3**(4), 261–270 (2002). DOI 10.1109/TITS.2002.804751

[17] Chen, C., Seff, A., Kornhauser, A., Xiao, J.: DeepDriving: Learning affordance for direct perception in autonomous driving. In: Proc. of the IEEE Int. Conf. on Computer Vision (ICCV), pp. 2722–2730 (2015)

[18] Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(8), 790–799 (1995). DOI 10.1109/34.400568

[19] Chiel, H.J., Beer, R.D.: The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. Trends in neurosciences **20**(12), 553–557 (1997)

[20] Cisek, P.: Beyond the computer metaphor: Behaviour as interaction. Journal of Consciousness Studies **6**(11-12), 125–142 (1999)

[21] Cliff, D., Husbands, P., Harvey, I.: Explorations in evolutionary robotics. Adaptive Behavior **2**(1), 73–110 (1993)

[22] Cox, D.D., Dean, T.: Neural networks and neuroscience-inspired computer vision. Current Biology **24**(18), R921 – R929 (2014). DOI http://dx.doi.org/10.1016/j.cub.2014.08.026. URL `http://www.sciencedirect.com/science/article/pii/S0960982214010392`

[23] Crisman, J.D., Thorpe, C.E.: Unscarf-a color vision system for the detection of unstructured roads. In: Proceedings. 1991 IEEE International Conference on Robotics and Automation, pp. 2496–2501 vol.3 (1991). DOI 10.1109/ROBOT.1991.132000

[24] Crisman, J.D., Thorpe, C.E.: Scarf: a color vision system that tracks roads and intersections. IEEE Transactions on Robotics and Automation **9**(1), 49–58 (1993). DOI 10.1109/70.210794

[25] de Croon, G.: Adaptive active vision. Ph.D. thesis, Maastricht University (2008)

[26] Dasgupta, D., Michalewicz, Z.: Evolutionary algorithms in engineering applications. Springer Science & Business Media (2013)

[27] Dauphin, Y., de Vries, H., Bengio, Y.: Equilibrated adaptive learning rates for non-convex optimization. In: Advances in Neural Information Processing Systems, pp. 1504–1512 (2015)

[28] Davies, B., Lienhart, R.: Using cart to segment road images. In: Electronic Imaging 2006, pp. 60,730U–60,730U. International Society for Optics and Photonics (2006)

[29] Dickmanns, E., Mysliwetz, B.: Recursive 3-D road and relative ego-state recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(2), 199–213 (1992)

[30] Dickmanns, E.D., Zapp, A.: Autonomous high speed road vehicle guidance by computer vision. In: International Federation of Automatic Control. World Congress (10th). Automatic control: world congress., vol. 1 (1988)

[31] Duarte, M., Oliveira, S.M., Christensen, A.L.: Evolution of hybrid robotic controllers for complex tasks. Journal of Intelligent & Robotic Systems **78**(3), 463–484 (2015). DOI 10.1007/s10846-014-0086-x. URL `https://doi.org/10.1007/s10846-014-0086-x`

[32] Dudek, G., Jenkin, M.: Computational Principles of Mobile Robotics. Cambridge University Press, New York, NY, USA (2000)

[33] Felleman, D.J., Van Essen, D.C.: Distributed hierarchical processing in the primate cerebral cortex. Cerebral cortex (New York, NY: 1991) **1**(1), 1–47 (1991)

[34] Floreano, D., Kato, T., Marocco, D., Sauser, E.: Coevolution of active vision and feature selection. Biological Cybernetics **90**(3), 218–228 (2004)

[35] Fogel, I., Sagi, D.: Gabor filters as texture discriminator. Biological cybernetics **61**(2), 103–113 (1989)

[36] Funahashi, K., Nakamura, Y.: Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks. Neural Networks **6**, 801–806 (1993)

[37] Gabriel, E., Fagg, G.E., Bosilca, G., Angskun, T., Dongarra, J.J., Squyres, J.M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., et al.: Open mpi: Goals, concept, and design of a next generation mpi implementation. In: European Parallel Virtual Machine/Message Passing Interface Users? Group Meeting, pp. 97–104. Springer (2004)

[38] Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)

[39] Gibson, J.J.: The ecological approach to visual perception: classic edition. Psychology Press (2014)

[40] Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 315–323 (2011)

[41] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA (1989)

[42] Hadsell, R., Sermanet, P., Ben, J., Erkan, A., Scoffier, M., Kavukcuoglu, K., Muller, U., LeCun, Y.: Learning long-range vision for autonomous off-road driving. Journal of Field Robotics **26**(2), 120–144 (2009)

[43] Harvey, I., Di Paolo, E., Wood, R., Quinn, M., Tuci, E.: Evolutionary robotics: A new scientific tool for studying cognition. Artificial Life **11**(2), 79–98 (2005)

[44] Harvey, I., Husbands, P., Cliff, D.: Seeing the light: artificial evolution, real vision. In: D. Cliff, P. Husbands, J. Meyer, S. Wilson (eds.) Proc. of $3^{rd}$ Int. Conf. on Simulation of Adaptive Behavior (SAB), pp. 392–401. MIT Press/Bradford Books, Boston MA (1994)

[45] Harvey, I., Husbands, P., Cliff, D., Thompson, A., Jakobi, N.: Evolutionary robotics: the sussex approach. Robotics and Autonomous Systems **20**(2–4), 205–224 (1997)

[46] Hawkins, D.M.: The problem of overfitting. Journal of chemical information and computer sciences **44**(1), 1–12 (2004)

[47] Haykin, S., Network, N.: A comprehensive foundation. Neural Networks **2**(2004), 41 (2004)

[48] Hel-Or, Y., Hel-Or, H.: Real-time pattern matching using projection kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(9), 1430–1445 (2005). DOI 10.1109/TPAMI.2005.184

[49] Held, R., Hein, A.: Movement-produced stimulation in the development of visually guided behavior. Journal of Comparative and Physiological Psychology **56**(5), 872–876 (1963)

[50] Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology **160**(1), 106–154 (1962)

[51] Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., et al.: An empirical evaluation of deep learning on highway driving. arXiv preprint arXiv:1504.01716 (2015)

[52] Jakobi, N.: Evolutionary robotics and the radical envelope-of-noise hypothesis. Adaptive behavior **6**(2), 325–368 (1997)

[53] Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: Computer Vision, 2009 IEEE 12th International Conference on, pp. 2146–2153 (2009). DOI 10.1109/ICCV.2009.5459469

[54] Jensen, J.R., et al.: Introductory digital image processing: a remote sensing perspective. Ed. 2. Prentice-Hall Inc. (1996)

[55] Jeong, H., Oh, Y., Park, J., Koo, B., Lee, S.: Vision-based adaptive and recursive tracking of unpaved roads. Pattern Recognition Letters **23**, 73 – 82 (2002)

[56] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)

[57] Joachims, T.: Svmlight: Support vector machine. SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund **19**(4) (1999)

[58] Jochem, T.M., Baluja, S.: A massively parallel road follower. In: 1993 Computer Architectures for Machine Perception, pp. 2–12 (1993). DOI 10.1109/CAMP.1993.622451

[59] Jochem, T.M., Pomerleau, D.A., Thorpe, C.E.: MANIAC: A next generation neurally based autonomous road follower. In: Proc. of the Int. Conf. on Intelligent Autonomous Systems (1993)

[60] Kato, T., Floreano, D.: An evolutionary active-vision system. In: Proc. of the Congress on Evolutionary Computation (CEC), vol. 1, pp. 107–114. IEEE (2001)

[61] Kim, D., Sun, J., Oh, S.M., Rehg, J.M., Bobick, A.F.: Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In: Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pp. 518–525. IEEE (2006)

[62] Kluge, K., Thorpe, C.: The YARF system for vision-based road following. Mathematical and Computer Modelling **22**(4-7), 213–233 (1995)

[63] Koutník, J., Cuccu, G., Schmidhuber, J., Gomez, F.: Evolving large-scale neural networks for vision-based reinforcement learning. In: Proceedings of the 15th annual conference on Genetic and evolutionary computation, pp. 1061–1068. ACM (2013)

[64] Koutník, J., Schmidhuber, J., Gomez, F.: Evolving deep unsupervised convolutional networks for vision-based reinforcement learning. In: Proceedings of the 2014 An-

nual Conference on Genetic and Evolutionary Computation, pp. 541–548. ACM (2014)

[65] Koutník, J., Schmidhuber, J., Gomez, F.: Online evolution of deep convolutional network for vision-based reinforcement learning. In: International Conference on Simulation of Adaptive Behavior, pp. 260–269. Springer (2014)

[66] Kriegeskorte, N.: Deep neural networks: A new framework for modeling biological vision and brain information processing. Annual Review of Vision Science **1**(1), 417–446 (2015). DOI 10.1146/annurev-vision-082114-035447. URL https://doi.org/10.1146/annurev-vision-082114-035447

[67] Kristensen, D.: Autonomous road following-a study of methods for tracking unmarking roads in image sequences sweden kth numerical analysis and computer. Science pp. 30–33 (2004)

[68] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)

[69] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012). URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[70] Le, Q.V., Monga, R., Devin, M., Corrado, G., Chen, K., Ranzato, M., Dean, J., Ng, A.Y.: Building high-level features using large scale unsupervised learning. CoRR **abs/1112.6209** (2011). URL http://arxiv.org/abs/1112.6209

[71] LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. In: M. Arbib (ed.) The handbook of brain theory and neural networks, vol. 3361, pp. 255–258. MIT Press (1995)

[72] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)

[73] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). DOI 10.1109/5.726791

[74] Lee, J., Crane, C.D.: Road following in an unstructured desert environment based on the em(expectation-maximization) algorithm. In: 2006 SICE-ICASE International Joint Conference, pp. 2969–2974 (2006). DOI 10.1109/SICE.2006.314963

[75] Liu, Z., Wang, Y.: Deeper direct perception in autonomous driving. Technical report (2016)

[76] Lohn, J., Hornby, G., Linden, D.: An evolved antenna for deployment on nasa?s space technology 5 mission. Genetic Programming Theory and Practice II pp. 301–315 (2005)

[77] Loose, H., Franke, U., Stiller, C.: Kalman particle filter for lane recognition on rural roads. In: 2009 IEEE Intelligent Vehicles Symposium, pp. 60–65 (2009). DOI 10.1109/IVS.2009.5164253

[78] Lowe, D.G.: Object recognition from local scale-invariant features. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2, pp. 1150–1157. Ieee (1999)

[79] Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)

[80] Manz, M., Himmelsbach, M., Luettel, T., Wuensche, H.J.: Fusing lidar and vision for autonomous dirt road following. In: Autonome Mobile Systeme 2009, pp. 17–24. Springer (2009)

[81] Manz, M., von Hundelshausen, F., Wuensche, H.J.: A hybrid estimation approach for autonomous dirt road following using multiple clothoid segments. In: 2010 IEEE International Conference on Robotics and Automation, pp. 2410–2415 (2010). DOI 10.1109/ROBOT.2010.5509983

[82] Meyer, J.A., Husbands, P., Harvey, I.: Evolutionary robotics: A survey of applications and problems. In: Evolutionary Robotics, pp. 1–21. Springer (1998)

[83] Moody, J., Darken, C.: Learning with localized receptive fields. Yale Univ., Department of Computer Science (1988)

[84] Moon, T.K.: The expectation-maximization algorithm. IEEE Signal Processing Magazine **13**(6), 47–60 (1996). DOI 10.1109/79.543975

[85] Nöe, A.: Action in Perception. MIT Press (2006)

[86] Nolfi, S., Bongard, J.C., Husbands, P., Floreano, D.: Evolutionary robotics. (2016)

[87] Öfjäll, K., Felsberg, M.: Biologically inspired online learning of visual autonomous driving. In: British Machine Vision Conference 2014, Nottingham, UK September 1-5 2014, pp. 137–156. BMVA Press (2014)

[88] Öfjäll, K., Felsberg, M.: Online learning of vision-based robot control during autonomous operation. In: New Development in Robot Vision, pp. 137–156. Springer (2015)

[89] Ofjall, K., Felsberg, M., Robinson, A.: Visual autonomous road following by symbiotic online learning. In: 2016 IEEE Intelligent Vehicles Symposium (IV), pp. 136–143 (2016). DOI 10.1109/IVS.2016.7535377

[90] Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision **42**(3), 145–175 (2001)

[91] Ososinski, M., Labrosse, F.: Automatic driving on ill-defined roads: An adaptive, shape-constrained, color-based method. Journal of Field Robotics **32**(4), 504–533 (2015)

[92] Paz, L.M., Piniés, P., Newman, P.: A variational approach to online road and path segmentation with monocular vision. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on, pp. 1633–1639. IEEE (2015)

[93] Peniak, M., Marocco, D., Ramirez-Contla, S., Cangelosi, A.: Active vision for navigating unknown environments: An evolutionary robotics approach for space research. In: ESA Special Publication, vol. 673 (2009)

[94] Pomerleau, D.: Progress in neural network-based vision for autonomous robot driving. In: Proc. of the Intelligent Vehicles '92 Symposium, pp. 391–396 (1992)

[95] Ramstrom, O., Christensen, H.: A method for following unmarked roads. In: Intelligent Vehicles Symposium, 2005. Proceedings. IEEE, pp. 650–655. IEEE (2005)

[96] Rasmussen, C.: Combining laser range, color, and texture cues for autonomous road following. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), vol. 4, pp. 4320–4325 vol.4 (2002). DOI 10.1109/ROBOT.2002.1014439

[97] Rasmussen, C.: Grouping dominant orientations for ill-structured road following. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., vol. 1, pp. I–470–I–477 Vol.1 (2004). DOI 10.1109/CVPR.2004.1315069

[98] Rosenblum, M.: Neurons that know how to drive. In: Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511), pp. 556–562 (2000). DOI 10.1109/IVS.2000.898406

[99] S. Thrun et al.: Stanley: The robot that won the DARPA grand challenge. Journal of Field Robotics **23**(9), 661–692 (2006)

[100] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)

[101] Shin, D.H., Singh, S.: Path generation for robot vehicles using composite clothoid segments. Tech. rep., DTIC Document (1990)

[102] Shinzato, P.Y., Wolf, D.F.: A road following approach using artificial neural networks combinations. Journal of Intelligent & Robotic Systems **62**(3), 527–546 (2011)

[103] Smuda, P., Schweiger, R., Neumann, H., Ritter, W.: Multiple cue data fusion with particle filters for road course detection in vision systems. In: 2006 IEEE Intelligent Vehicles Symposium, pp. 400–405 (2006). DOI 10.1109/IVS.2006.1689661

[104] Soquet, N., Aubert, D., Hautiere, N.: Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation. In: 2007 IEEE Intelligent Vehicles Symposium, pp. 160–165 (2007). DOI 10.1109/IVS.2007.4290108

[105] Southall, B., Taylor, C.J.: Stochastic road shape estimation. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 1, pp. 205–212 vol.1 (2001). DOI 10.1109/ICCV.2001.937519

[106] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1), 1929–1958 (2014)

[107] Suzuki, M., Floreano, D.: Evolutionary Active Vision Toward Three Dimensional Landmark-Navigation, pp. 263–273. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). DOI 10.1007/11840541_22

[108] Suzuki, M., Floreano, D.: Enactive robot vision. Adaptive Behavior **16**(2-3), 122–128 (2008)

[109] Suzuki, M., Floreano, D., Paolo, D., Ezequiel, A.: The contribution of active body movement to visual development in evolutionary robots. Neural Networks **18**(5-6), 656–665 (2005)

[110] Suzuki, M., Floreano, D., Paolo, E.A.D.: Constraints on body movement during visual development affect behavior of evolutionary robots. In: Proc. of the Int. Joint Conf. on Neural Networks (2005)

[111] Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1701–1708 (2014)

[112] Tan, C., Hong, T., Chang, T., Shneier, M.: Color model-based real-time learning for road following. In: 2006 IEEE Intelligent Transportation Systems Conference, pp. 939–944 (2006). DOI 10.1109/ITSC.2006.1706865

[113] Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints **abs/1605.02688** (2016). URL `http://arxiv.org/abs/1605.02688`

[114] Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 1521–1528. IEEE (2011)

[115] Tuci, E.: An Exploration on the Evolution of Learning Behaviour Using Robot-based Models. University of Sussex (2004)

[116] Tudoran, C.T., Neagoe, V.E.: A new neural network approach for visual autonomous road following. Latest Trends on Computers **1**, 266–271 (2010)

[117] Valada, A., Oliveira, G., Brox, T., Burgard, W.: Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In: Proc. of the Int. Symposium on Experimental Robotics, pp. 465–477 (2016)

[118] Vitor, G.B., Lima, D.A., Victorino, A.C., Ferreira, J.V.: A 2d/3d vision based approach applied to road detection in urban environments. In: 2013 IEEE Intelligent Vehicles Symposium (IV), pp. 952–957 (2013). DOI 10.1109/IVS.2013.6629589

[119] Vosselman, G., de Knecht, J.: Road tracing by profile matching and kaiman filtering. In: Automatic Extraction of Man-Made Objects from Aerial and Space Images, pp. 265–274. Springer (1995)

[120] Wymann, B., Espié, E., Guionneau, C., Dimitrakakis, C., Coulom, R., Sumner, A.: Torcs, the open racing car simulator. Software available at http://torcs. sourceforge. net (2000)

[121] Yeo, Y., Xiao, X., Zhang, X.: Rural scene parsing and road boundary estimation by fusion of lidar pointcloud and eo images. In: 2016 19th International Conference on Information Fusion (FUSION), pp. 1760–1767 (2016)

[122] Yuan, Y., Jiang, Z., Wang, Q.: Video-based road detection via online structural learning. Neurocomputing **168**, 336 – 347 (2015). DOI https://doi.org/10.1016/j.neucom.2015.05.092. URL http://www.sciencedirect.com/science/article/pii/S0925231215007900

[123] Zhang, A.M., Kleeman, L.: A panoramic color vision system for following ill-structured roads (2006)

[124] Zhou, S., Gong, J., Xiong, G., Chen, H., Iagnemma, K.: Road detection using support vector machine based on online learning and evaluation. In: Intelligent Vehicles Symposium (IV), 2010 IEEE, pp. 256–261. IEEE (2010)